

一套配教学视频的项目开发案例类丛书，累计销售160000册



12套
完整项目
开发案例

网站、QQ等技术支持，让学习畅通无阻

10小时
教学视频





模块库、题库、素材库登录本书网站



于国槐 王雨竹 编著

PHP
典型模块开发全程实录

- 注册登录模块（视频讲解：45分）
- 网页计数器模块（视频讲解：50分）
- 上传与下载模块（视频讲解：38分）
- FTP管理系统（视频讲解：8分）
- 假日公告管理模块（视频讲解：12分）
- 分页显示模块（视频讲解：48分）
- 论坛模块（视频讲解：4小时22分）
- 在线支付模块（视频讲解：24分）
- 留言本模块（视频讲解：30分）
- 博客模块（视频讲解：14分）
- 创建PDF文档模块（视频讲解：30分）
- 邮件发送系统（视频讲解：45分）

清華大學出版社

软件项目开发全程实录

PHP 典型模块开发全程实录

于国槐　王雨竹　编著

清華大學出版社
北　京

内 容 简 介

《PHP 典型模块开发全程实录》以注册登录模块、网页计数器模块、上传与下载模块、FTP 管理系统、假日公告管理模块、分页显示模块、论坛模块、在线支付模块、留言本模块、博客模块、创建 PDF 文档模块、邮件发送系统 12 个实际模块开发程序为案例，从软件工程的角度出发，按照模块的开发顺序，系统、全面地介绍了程序开发流程。从开发背景、需求分析、系统功能分析、数据库分析、数据库建模、模块开发到模块的编译与发布，对每一过程都进行了详细的介绍。

本书及光盘特色还有：12 套项目开发完整案例，项目开发案例的同步视频和其源程序。登录网站还可获取各类资源库（模块库、题库、素材库）等项目案例常用资源，网站还提供技术论坛支持等。

本书案例涉及行业广泛，实用性非常强。通过本书的学习，读者可以了解各个行业的特点，能够针对某一行业进行软件开发，也可以通过光盘中提供的案例源代码和数据库进行二次开发，以减少开发系统所需要的时间。

本书适合各级 PHP 网站开发人员学习使用，也可供大、中专院校师生学习参考。

图书在版编目（CIP）数据

PHP 典型模块开发全程实录/于国槐，王雨竹编著．—北京：清华大学出版社，2013
（软件项目开发全程实录）

ISBN 978-7-302-33740-9

I．①P…　II．①软…　III．①PHP 语言–程序设计　IV．①TP312

中国版本图书馆 CIP 数据核字（2013）第 204482 号

责任编辑： 赵洛育
封面设计： 陈　敏
版式设计： 文森时代
责任校对： 李虎斌
责任印制：
出版发行： 清华大学出版社
　网　　址： http://www.tup.com.cn，http://www.wqbook.com
　地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
　社 总 机： 010-62770175　　**邮　　购：** 010-62786544
　投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
　质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 203mm×260mm　**印　　张：** 21.5　**字　　数：** 605 千字
（附 DVD 光盘 1 张）
版　　次： 2013 年 10 月第 1 版　**印　　次：** 2013 年 10 月第 1 次印刷
印　　数： 1～4000
定　　价： 59.80 元

产品编号：051399-01

前　言

Preface

编写目的与背景

众所周知，当前社会需求和高校课程设置严重脱节，一方面企业找不到可迅速上手的人才，另一方面大学生就业难。如果有一些面向工作应用的案例参考书，让大学生得以参考，并能亲手去做，势必能缓解这种矛盾。本书就是这样一本书：项目开发案例型的、面向工作应用的软件开发类图书。编写本书的首要目的就是架起让学生从学校走向社会的桥梁。

其次，本书以完成小型项目为目的，让学生切身感受到软件开发给工作带来实实在在的用处和方便，并非只是枯燥的语法和陌生的术语，从而激发学生学习软件的兴趣，让学生变被动学习为自主自发学习。

第三，本书的项目开发案例过程完整，不但适合在学习软件开发时作为小型项目开发的参考书，而且可以作为毕业设计的案例参考书。

第四，从书第 1 版于 2008 年 6 月出版，于 2011 年 1 月改版，因为编写细腻，配备全程视频备受读者瞩目，丛书累计销售 16 万册，成为近年来最受欢迎的软件开发项目案例类丛书之一。

在以上背景下，我们根据读者朋友的反馈，与时俱进，对丛书进行了改版。

本书特点

视频讲解

对于初学者来说，视频讲解是最好的导师，它能够引导初学者快速入门，使初学者感受到编程的快乐和成就感，增强进一步学习的信心。鉴于此，本书为每一个案例都配备了视频讲解，初学者可以通过视频讲解实现案例中的功能。

典型案例

本书案例均从实际应用角度出发，应用了当前流行的技术，涉及的知识广泛，读者可以从每个案例中积累丰富的实战经验。

代码注释

为了便于读者阅读程序代码，书中的代码均提供了详细的注释，并且整齐地纵向排列，可使读者快速领略作者意图。

代码贴士

案例类书籍通常会包含大量的程序代码，冗长的代码往往令初学者望而生畏。为了方便读者阅读和理解代码，本书避免了连续大篇幅的代码，将其分割为多个部分，并对重要的变量、方法和知识点设计了独具特色的代码贴士。

知识扩展

为了增加读者的编程经验和技巧，书中每个案例都标记有注意、技巧等提示信息，并且在每章中都提供有一项专题技术。

本书约定

由于篇幅有限，本书每章并不能逐一介绍案例中的各模块。作者选择了基础和典型的模块进行介绍，对于功能重复的模块，由于技术、设计思路和实现过程基本雷同，因此没有在书中体现。读者在学习过程中若有相关疑问，请登录本书官方网站。本书中涉及的功能模块在光盘中都附带有视频录像，方便读者学习。

适合读者

本书适合作为计算机相关专业的大学生、软件开发相关求职者和爱好者的毕业设计和项目开发的参考书。

本书服务

为了给读者提供更为方便快捷的服务，读者可以登录本书官方网站：www.rjkflm.com，或者加入QQ：4006751066 进行交流。

本书作者

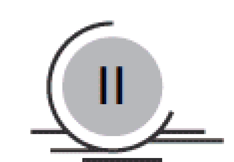
本书由明日科技软件开发团队组织编写，主要由于国槐、王雨竹执笔，如下人员也参与了本书的编写工作，他们是：辛洪郁、杨丽、张鑫、王小科、王国辉、顾彦玲、赛奎春、高春艳、陈英、宋禹蒙、刘佳、刘莉莉、陈丹丹、隋光宇、郭鑫、刘志铭、李伟、张金辉、李慧、刘欣、李继业、潘凯华、赵永发、寇长梅、赵会东、王敬洁、李浩然、苗春义、刘清怀、张世辉、张领等，在此一并感谢！

在编写本书的过程中，我们本着科学、严谨的态度，力求精益求精，但错误、疏漏之处在所难免，敬请广大读者批评指正。

感谢您购买本书，希望本书能成为您的良师益友，成为您步入编程高手之路的踏脚石。

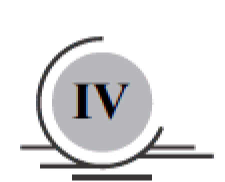
宝剑锋从磨砺出，梅花香自苦寒来。祝读书快乐！

编　　者



目 录

Contents





第 1 章

注册登录模块

（PHP+Ajax+Zend_Mail 实现）

从动态网站开始，注册登录模块就成为了网站必不可少的组成部分，因为其功能单一，实现相对简单，从而成为入门级的模块。随着 Web 2.0 的发展，人机交互成为主流趋势，注册登录模块也顺应了时代的发展，变得更加有“亲和力”。像过去那种花了许久时间填写注册表单，提交后发现信息填写错误，需要重新填写的惨痛经历越来越少。随着 Ajax 技术的兴起，这种情况就更少发生了。本章将开发一个功能齐全并有着良好交互性的注册登录模块。

通过阅读本章，可以学习到：

- 注册信息实时提示
- E-mail 激活技术
- GD2 验证码
- 更换验证码
- 防 SQL 注入技术

1.1 注册及登录模块概述

注册及登录模块的功能比较单一，除了实现验证用户信息的有效性和一些必要的安全设置外，更多的是如何提高程序的可操作性，更好地方便用户。注册及登录模块由用户注册、用户登录和找回密码 3 部分组成。

1.1.1 用户注册流程

用户注册的信息分为必填信息和可选信息。必填信息默认是可见的，而可选信息默认是隐藏的，在必填信息不完整时，“注册”按钮是不可用的。对于可选信息，用户可以选择填写或不填写。但如果填写了，那么就需要对信息的正确性进行检验。当用户注册成功后，系统将向用户所填写的 E-mail 发送一封激活邮件，用户只有激活后，才可使用该账号进行登录。用户注册流程图如图 1.1 所示。

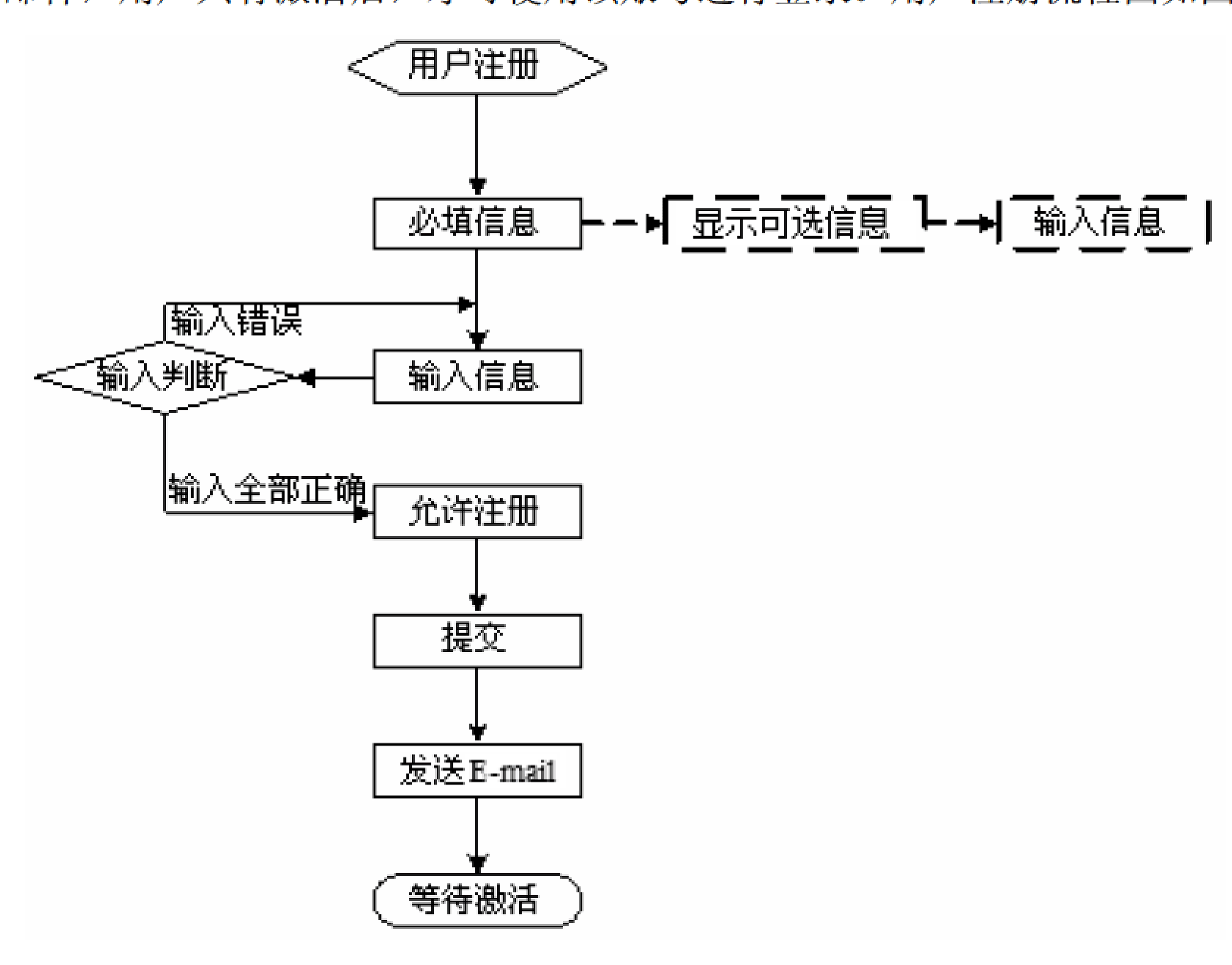
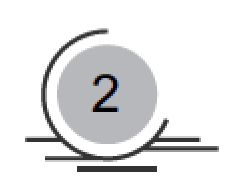


图 1.1　用户注册流程图

1.1.2 用户登录流程

用户登录时，除了用户名和密码外，还需要填写随机生成的验证码。当验证码显示不清楚时，可以通过超链接进行更换而无须重新刷新页面。登录处理对登录进行了一些限制，例如，输入 3 次无效用户名后，将使用 Cookie 技术禁止该用户继续登录；连续 3 次输入错误密码后，该账号将被冻结。用户登录流程图如图 1.2 所示。





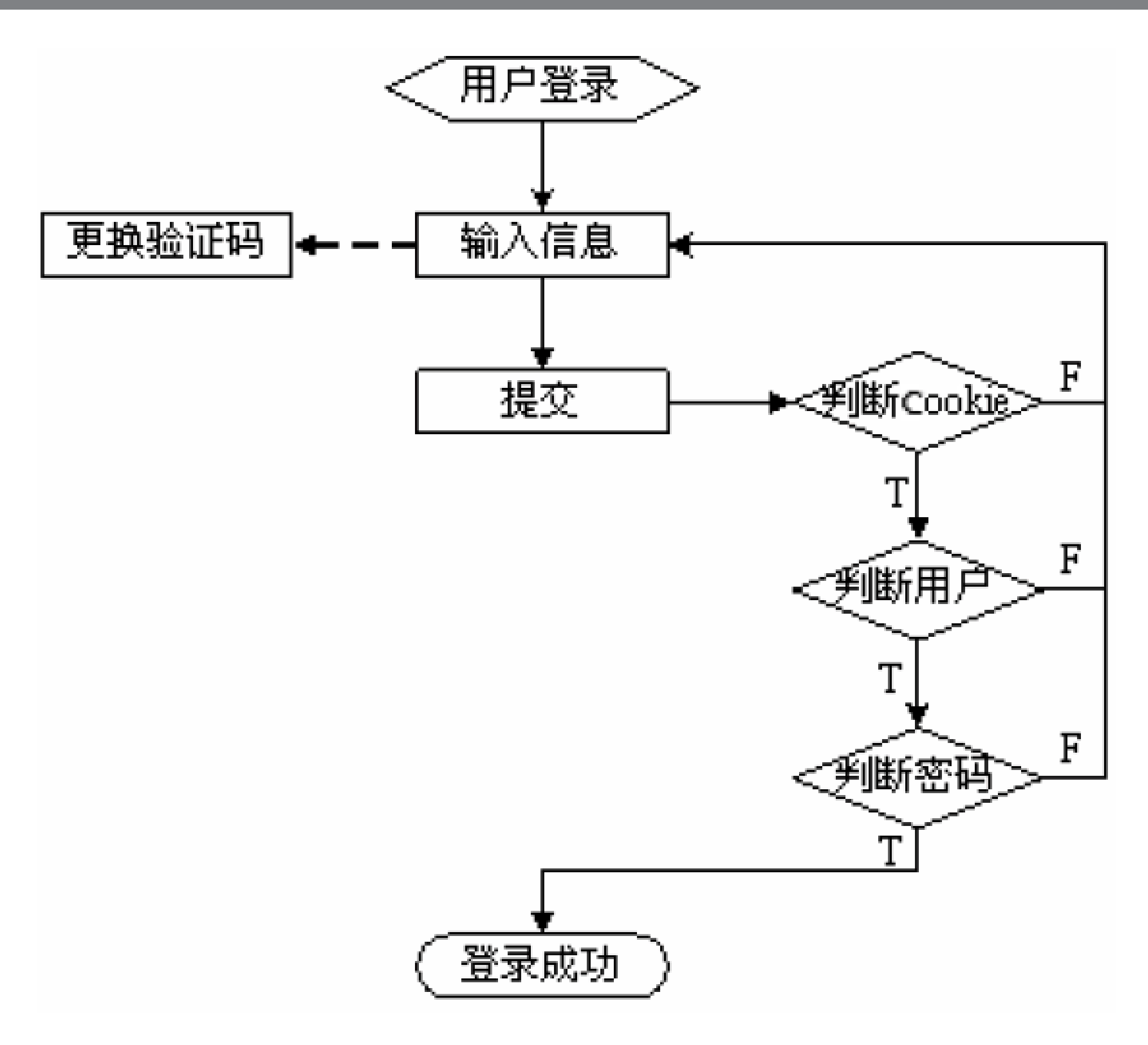


图 1.2　用户登录流程图

1.1.3　找回密码流程

当用户因为个别原因忘记密码时，可以通过密码保护问题及答案来找回密码。如果用户输入正确，那么系统将密码通过电子邮件的形式发到用户的注册邮箱中。如果没有填写密码保护的问题及答案或者回答错误，则无法找回。找回密码流程图如图 1.3 所示。

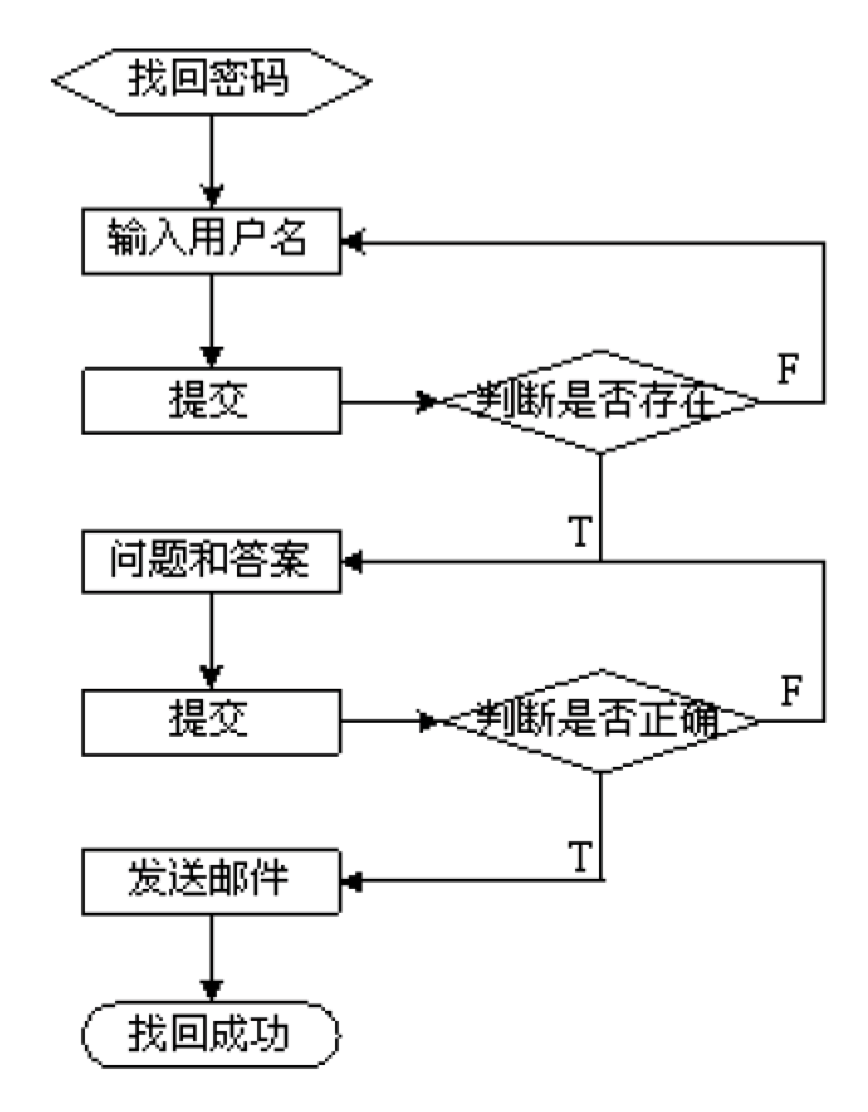


图 1.3　找回密码流程图

1.1.4　程序预览

注册登录模块主要由两个界面组成，即注册页面和登录页面。注册页面与登录页面的运行效果如

图 1.4 和图 1.5 所示。



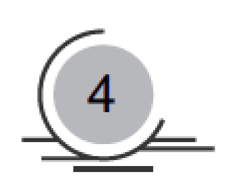
图 1.4　注册页面的运行效果



图 1.5　登录页面的运行效果

1.2　数据库设计

本模块应用 MySQL 数据库存储用户注册信息，有关 MySQL 数据库及数据表的创建将在本节进行



详细讲解。

1.2.1　数据库设计概述

注册及登录验证模块使用的数据库是 db_reglog，库中包含了一个数据表 tb_member，该表中有 12 个字段。tb_member 表结构及说明如图 1.6 所示。

服务器: localhost ▸ 数据库: db_reglog ▸ 表 : tb_member

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

	字段	类型	整理	属性	Null	默认	额外	说明
☐	id	int(4)			否		auto_increment	自动编号
☐	name	varchar(100)	gb2312_chinese_ci		否			用户名称
☐	password	varchar(200)	gb2312_chinese_ci		否			用户密码
☐	answer	varchar(100)	gb2312_chinese_ci		否			密保答案
☐	question	varchar(100)	gb2312_chinese_ci		否			密保问题
☐	email	varchar(100)	gb2312_chinese_ci		否			Email
☐	realname	varchar(200)	gb2312_chinese_ci		否			登录限制
☐	birthday	date			否			出生日期
☐	telephone	varchar(20)	gb2312_chinese_ci		否			电话号码
☐	qq	varchar(15)	gb2312_chinese_ci		否			QQ号
☐	count	int(1)			否	0		登录限制
☐	active	int(1)			否	0		是否激活

图 1.6　tb_member 表结构及说明

1.2.2　封装数据库连接-操作类

为了便于维护，减少代码冗余，本模块使用了一个简单的自定义数据库类。下面给出类中的主要代码。

（1）类中的成员变量。其中$host、$name、$pwd 和$dBase 变量是类初始化时所使用的，有默认值，其他变量都是相关的数据信息。代码如下：

```
<?php
class opmysql{
private $host = 'localhost';                    //服务器地址
private $name = 'root';                         //登录账号
private $pwd = '111';                           //登录密码
private $dBase = 'db_reglog';                   //数据库名称
private $conn = '';                             //数据库连接资源
private $result = '';                           //结果集
private $msg = '';                              //返回结果
private $fields;                                //返回字段
private $fieldsNum = 0;                         //返回字段数
private $rowsNum = 0;                           //返回结果数
private $rowsRst = '';                          //返回单条记录的字段数组
private $filesArray = array();                  //返回字段数组
private $rowsArray = array();                   //返回结果数组
```

（2）构造函数。该构造函数有 4 个参数，这 4 个参数都可以被省略而使用默认值。参数处理后，调用 init_conn()函数。代码如下：

```
function __construct($host='',$name='',$pwd='',$dBase=''){
if($host != '')
$this->host = $host;
if($name != '')
$this->name = $name;
if($pwd != '')
$this->pwd = $pwd;
if($dBase != '')
$this->dBase = $dBase;
$this->init_conn();
}
```

init_conn()函数根据成员变量中的值来创建数据库连接源。代码如下：

```
//连接数据库
function init_conn(){
$this->conn=@mysql_connect($this->host,$this->name,$this->pwd);
@mysql_select_db($this->dBase,$this->conn);
mysql_query("set names gb2312");                          //设置编码
}
```

（3）类中的操作函数。

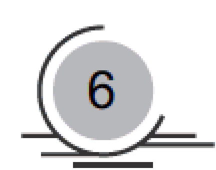
① 查询函数。查询函数根据传过来的 SQL 语句进行查询，并将查询结果保存到成员变量$result 中。符号“@”的作用是屏蔽错误信息。代码如下：

```
//查询结果
function mysql_query_rst($sql){
if($this->conn == ''){
$this->init_conn();
}
$this->result = @mysql_query($sql,$this->conn);
}
```

② 返回查询记录数函数。该函数根据查询结果返回记录数。代码如下：

```
function getRowsNum($sql){
$this->mysql_query_rst($sql);
if(mysql_errno() == 0){
return @mysql_num_rows($this->result);
}else{
return '';
}
}
```

③ 取得记录数组函数。该函数将查询结果输出成一个数组并返回，其处理的是单条记录。代码如下：



```
function getRowsRst($sql){
$this->mysql_query_rst($sql);
if(mysql_error() == 0){
$this->rowsRst = mysql_fetch_array($this->result,MYSQL_ASSOC);
return $this->rowsRst;
}else{
return '';
}
}
```

④ 取得记录数组函数。功能同上，只是这里返回一个有多条记录的二维数组。代码如下：

```
//取得记录数组（多条记录）
function getRowsArray($sql){
$this->mysql_query_rst($sql);
if(mysql_errno() == 0){
while($row = mysql_fetch_array($this->result,MYSQL_ASSOC)) {
$this->rowsArray[] = $row;
}
return $this->rowsArray;
}else{
return '';
}
}
```

⑤ 返回更新、删除、添加的记录数函数。该函数用来更新、删除、添加记录并获取受影响的记录数，用来判断操作是否成功。代码如下：

```
function uidRst($sql){
if($this->conn == ''){
$this->init_conn();
}
@mysql_query($sql);
$this->rowsNum = @mysql_affected_rows();
if(mysql_errno() == 0){
return $this->rowsNum;
}else{
return '';
}
}
```

⑥ 释放结果集函数。该函数用来将不再使用的数据删除，释放内存。代码如下：

```
function close_rst(){
mysql_free_result($this->result);
$this->msg = '';
$this->fieldsNum = 0;
$this->rowsNum = 0;
$this->filesArray = '';
```

```
$this->rowsArray = '';
}
```

⑦ 关闭数据库函数。代码如下：

```
//关闭数据库
function close_conn(){
$this->close_rst();
mysql_close($this->conn);
$this->conn = '';
}
}
$conne = new opmysql();
?>
```

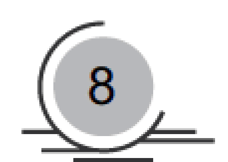
1.3　用 户 注 册

1.3.1　用户注册功能概述

在用户注册时，需要填写的必填信息包括用户名、密码和 E-mail。只有这些必填信息填写完整并正确时，“注册”按钮才被激活。这时用户可以选择注册或填写详细信息后再进行注册。在整个注册的过程中，用户的每一次操作，系统都将给出友好的提示来帮助用户完成注册。注册页面的运行结果如图 1.7 所示。



图 1.7　注册页面的运行结果



用户注册功能包含 5 个核心文件，分别是 register.php（注册表单页）、register_chk.php（注册处理页）、chkname.php（检测用户名）、register.js（注册信息验证脚本）和 activation.php（激活处理页）。

1.3.2 注册页设计

注册表单页（register.php）中，主要包含填写信息的文本框、信息提示的 div 标签和 3 个相关按钮。注册页面中使用的元素如表 1.1 所示。

表 1.1 注册页面中使用的元素

元 素 名 称	元素属性值	说 明
script	"js/xmlhttprequest.js"	载入 xmlHttpRequest 对象脚本
script	"js/register.js"	载入注册信息验证脚本
text	id="regname" name="regname"	注册名称
div	id="namediv"	注册名称提示信息
password	id="regpwd1" name="regpwd1"	注册密码
div	id="pwddiv1"	密码提示信息
password	id="regpwd2" name="regpwd2"	确认密码
div	id="pwddiv2"	确认密码提示信息
text	id="email" name="email"	电子邮箱
div	id="emaildiv"	E-mail 提示信息
div	id="morediv" style="display:none;"	隐藏/显示非必填信息
text	id=" question" name="question"	密码保护问题
text	id="answer" name="answer"	密码保护答案
text	id="realname" name="realname"	真实姓名
text	id="birthday" name="birthday"	出生日期
text	id="telephone" name="telephone"	联系电话
text	id="qq "name=" qq"	QQ 号码
button	id="regbtn" disabled="disabled"	“注册”按钮
button	id="morebtn"	显示/隐藏全部注册信息
button	id="logbtn"	返回登录页面
div	id="imgdiv" style=" visibility: hidden;"	等待进度条

当用户向注册页面中的表单元素输入信息时，将会触发信息处理脚本 register.js。下面逐行讲解 register.js 脚本文件。

（1）在 register.js 中，首先写了一个通用函数，来避免重复输入 document.getElementById('…')。这个函数的意思是返回被触发的“id=idValue”元素对象。

```
function $(idValue){
	return document.getElementById(idValue);
}
```

（2）window.onload 事件，表示当窗体被载入时触发。function(){…}表示当页面被载入时所要进行的操作。

```
window.onload = function(){
	…
}
```

（3）下面所有的代码都是上面的函数。

① 将焦点定位到注册名称文本框，方便用户操作。接下来声明 5 个变量，这 5 个变量代表 5 项要检测的数据结果。当检测数据为“合格”时，代表它的变量将等于 yes。代码如下：

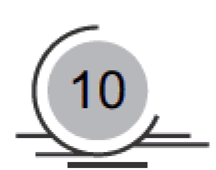
```
$('regname').focus();                          //焦点定位到注册名称文本框
//声明 5 个变量，表示要检测的 5 项数据
var cname1,cname2,cpwd1,cpwd2,cemail;
```

② chkreg()函数是每一次触发键盘事件后都要调用的。该函数判断 5 个变量的值，只有当所有变量都等于 yes 时，“注册”按钮才会被激活。代码如下：

```
function chkreg(){
if((cname1 == 'yes') && (cname2 == 'yes') && (cpwd1 == 'yes') && (cpwd2 == 'yes') && (cemail == 'yes')){
$('regbtn').disabled = false;
}else{
$('regbtn').disabled = true;
}
}
```

③ 验证用户名。当用户输入注册名称时，该函数会把用户的每次输入都做一下正则判断，并根据结果设置不同的 cname1 值。代码如下：

```
//验证用户名
$('regname').onkeyup = function (){
//获取注册名称
name = $('regname').value;
cname2 = '';
if(name.match(/^[a-zA-Z_]*/) == ''){
$('namediv').innerHTML = '<font color=red>必须以字母或下划线开头</font>';
cname1 = '';
}else if(name.length <= 3){
$('namediv').innerHTML = '<font color=red>注册名称必须大于 3 位</font>';
cname1 = '';
}else{
$('namediv').innerHTML = '<font color=green>注册名称符合标准</font>';
cname1 = 'yes';
}
chkreg();
}
```





④ 当注册名称文本框失去焦点时，即用户输入完毕转到页面中其他元素时，将检测用户名是否重复。用户名判断使用 Ajax 技术调用了 chkname.php 页（chkname.php 页的代码将在 1.3.3 节给出）并根据 chkname.php 的返回值，在 div 标签中显示判断结果。代码如下：

```
//当用户名文本框失去焦点时，系统调用处理页查看是否存在该用户
$('regname').onblur = function(){
name = $('regname').value;                                    //获取注册名称
//当用户名称的格式输入合格后，才进行这一步操作
if(cname1 == 'yes'){
xmlhttp.open('get','chkname.php?name='+name,true);            //创建新请求
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readyState == 4){
if(xmlhttp.status == 200){
var msg = xmlhttp.responseText;                               //获取响应页内容
if(msg == '1'){
$('namediv').innerHTML="<font color=green>恭喜您，该用户名可以使用!</font>";
cname2 = 'yes';
}else if(msg == '2'){
$('namediv').innerHTML="<font color=red>用户名被占用!</font>";
cname2 = '';
}else{
$('namediv').innerHTML="<font color=red>"+msg+"</font>";
cname2 = '';
}
}
}
chkreg();                                                     //检查是否要激活“注册”按钮
}
xmlhttp.send(null);
}
}
```

⑤ 验证密码。验证密码时，除了可以限制密码的长度外，还可以判断密码的强度。例如下面的代码中，6～12 位之间的密码都是弱，如果大于 12，但密码单一，如都是数字或文字，那么密码强度就为中，否则强度就为高。代码如下：

```
//验证密码
$('regpwd1').onkeyup = function(){
pwd = $('regpwd1').value;
pwd2 = $('regpwd2').value;
if(pwd.length < 6){
$('pwddiv1').innerHTML = '<font color=red>密码长度最少需要 6 位</font>';
cpwd1 = '';
}else if(pwd.length >= 6 && pwd.length < 12){
$('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：弱</font>';
cpwd1 = 'yes';
```

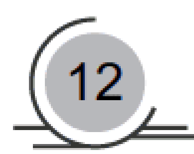
```
}else if((pwd.match(/^[0-9]*$/)!=null) || (pwd.match(/^[a-zA-Z]*$/) != null )){
$('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：中</font>';
cpwd1 = 'yes';
}else{
$('pwddiv1').innerHTML = '<font color=green>密码符合要求。密码强度：高</font>';
cpwd1 = 'yes';
}
if(pwd2 != '' && pwd != pwd2){
$('pwddiv2').innerHTML = '<font color=red>两次密码不一致!</font>';
cpwd2 = '';
}else if(pwd2 != '' && pwd == pwd2){
  $('pwddiv2').innerHTML = '<font color=green>密码输入正确</font>';
cpwd2 = 'yes';
}
chkreg();
}
```

⑥ 二次密码的验证比较简单，只要第一、第二次密码相等就可以。代码如下：

```
//验证二次密码
$('regpwd2').onkeyup = function(){
pwd1 = $('regpwd1').value;
pwd2 = $('regpwd2').value;
if(pwd1 != pwd2){
$('pwddiv2').innerHTML = '<font color=red>两次密码不一致!</font>';
cpwd2 = '';
}else{
$('pwddiv2').innerHTML = '<font color=green>密码输入正确</font>';
cpwd2 = 'yes';
chkreg();
}
}
```

⑦ E-mail 验证主要是对 E-mail 的格式进行检查，输入的字串中必须包含“@”和“.”，同时，这两个符号的位置既不能在首和尾，也不能连在一起。代码如下：

```
//验证 E-mail
$('email').onkeyup = function(){
emailreg = /^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$/;
$('email').value.match(emailreg);
if($('email').value.match(emailreg) == null){
$('emaildiv').innerHTML = '<font color=red>错误的 E-mail 格式</font>';
cemail = '';
}else{
$('emaildiv').innerHTML = '<font color=green>输入正确</font>';
cemail = 'yes';
}
```



```
chkreg();
}
```

⑧ 上面的部分是对必填项的检查。如果用户希望填写更详细的资料，可以单击“详细资料”按钮。代码如下：

```
//显示/隐藏详细信息
$('morebtn').onclick = function(){
if($('morediv').style.display == ''){
   $('morediv').style.display = 'none';
}else{
$('morediv').style.display = '';
}
}
```

⑨ 所有信息填写完毕并通过验证后，就可以注册了。用户单击“注册”按钮时，该段代码应用 Ajax 技术将获取的用户信息以 URL 的形式传给 register_chk.php 页，并根据返回结果做出相应的处理。代码如下：

```
//正式注册
$('regbtn').onclick = function(){
$('imgdiv').style.visibility = 'visible';
url = 'register_chk.php?name='+$('regname').value+'&pwd='+$('regpwd1').value+'&email='+$('email').value;
url += '&question=' +$('question').value+'&answer='+$('answer').value;
url += '&realname=' +$('realname').value+'&birthday='+$('birthday').value;
url += '&telephone='+$('telephone').value+'&qq='+$('qq').value;
alert(url);
return false;
xmlhttp.open('get',url,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readyState == 4){
if(xmlhttp.status == 200){
msg = xmlhttp.responseText;
if(msg == '1'){
alert('注册成功，请到您的邮箱中获取激活码！');
location='main.php';
}else if(msg == '-1'){
alert('您的服务器不支持 POP3，请仔细检查！！');
}else{
alert(msg);
}
}
$('imgdiv').style.visibility = 'hidden';
}
}
xmlhttp.send(null);
}
```

1.3.3 检测用户名是否被占用

检测用户名是否被占用使用的是 chkname.php 文件。其作用是检查用户名是否重复，并且返回判断结果。该页的代码如下：

```
<?php
//载入数据库类
include_once "conn/conn.php";
//根据 name 查询 tb_member 表
$sql = "select * from tb_member where name='".$_GET['name']."'";
//获取查询结果的记录数
$num = $conne->getRowsNum($sql);
if($num == 1){                                  //如果有记录
echo '2';
}else if($num == 0){                            //如果没有记录
echo '1';
}else{
echo $conne->msg_error();                       //否则出错
}
?>
```

1.3.4 注册信息处理

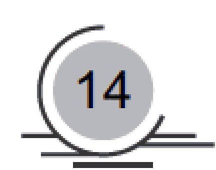
在注册信息处理页（register_chk.php）中，将 JavaScript 传递过来的值连成 insert 语句并执行。如果添加成功，则使用邮件函数发送一封激活邮件，并弹出提示框，结果如图 1.8 所示。



图 1.8 注册成功提示框

在处理代码中包含两种邮件的发送方法：一种是使用局域网，向指定 SMTP 服务器的邮箱中发送邮件，其中的 cym3100@163.com 邮箱是在 SMTP 服务器中注册的测试邮箱；另一种适用于本地网络，通过指定的邮箱（mrsoft@mrsoft.com 是在 WinMail 软件中注册的邮箱，用户名是 mrsoft@mrsoft.com，密码是 111）向用户提供的有效邮箱中发送邮件。处理页代码如下：

```
<?php
include_once 'conn/conn.php';
require_once 'Zend/Mail.php';                   //调用发送邮件的文件
require_once 'Zend/Mail/Transport/Smtp.php';    //调用 SMTP 验证文件
$reback = '0';
```



```
$url = 'http://'.$_SERVER['SERVER_NAME']. ':82'.dirname($_SERVER['SCRIPT_NAME']).'/activation.php';
$url .= '?name='.trim($_GET['name']).'&pwd='.md5(trim($_GET['pwd']));
//发送激活邮件
$subject="激活码的获取";
$mailbody='注册成功。您的激活码是：
'.'<a href="'.$url.'" target="_blank">'.$url.'</a><br>'.'请点击该地址，激活您的用户！';
//定义邮件内容
$envelope=" mrsoft@mrsoft.com ";                          //定义登录使用的邮箱
/*SMTP 测试版发送邮件方式，使用 SMTP 作为服务器
$tr = new Zend_Mail_Transport_Smtp('192.168.1.247');
$mail = new Zend_Mail();
$mail->addTo('cym3100@163.com','获取用户注册激活码');
$mail->setFrom('cym3100@163.com','明日科技典型模块程序测试邮箱，恭喜您用户注册成功！');
$mail->setSubject('获取注册用户的激活码');
$mail->setBodyHtml($mailbody);
$mail->send($tr);
*/
/*本地网络版发送邮件方法*/
$config = array('auth' => 'login',
'username' => 'mrsoft@mrsoft.com',
'password' => ' mingri');                                 //定义 SMTP 的验证参数
$transport = new Zend_Mail_Transport_Smtp('localhost', $config);          //实例化验证的对象
$mail = new Zend_Mail('GBK');                             //实例化发送邮件对象
$mail->setBodyHtml($mailbody);                            //发送邮件主体
$mail->setFrom($envelope, '明日科技典型模块程序测试邮箱，恭喜您用户注册成功！');//定义邮件发送使用的邮箱
$mail->addTo($_GET[email], '获取用户注册激活码');          //定义邮件的接收邮箱
$mail->setSubject('获取注册用户的激活码');                  //定义邮件主题
$mail->send($transport);                                  //执行发送操作
/*网络版发送邮件方法*/
$sql = "insert into tb_member(name,password,question,answer,email,realname,birthday,telephone,qq)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."','".$_GET['answer']."',
'".$_GET['email']."','".$_GET['realname']."','".$_GET['birthday']."','".$_GET['telephone']."','".$_GET['qq']."')";
$num = $conne->uidRst($sql);
if($num == 1){
$reback = '1';
}
echo $reback;
?>
```

1.3.5　注册用户激活

在用户注册成功后，需要登录用户填写的邮箱，浏览注册时发送的邮件，并且单击超链接，链接到 activation.php 文件，根据超链接传递的用户名和密码完成对指定用户表中 active 字段值的更新操作，即完成注册用户的激活操作。

登录 mrsoft@mrsoft.com 邮箱，将查看到如图 1.9 所示的邮件内容。

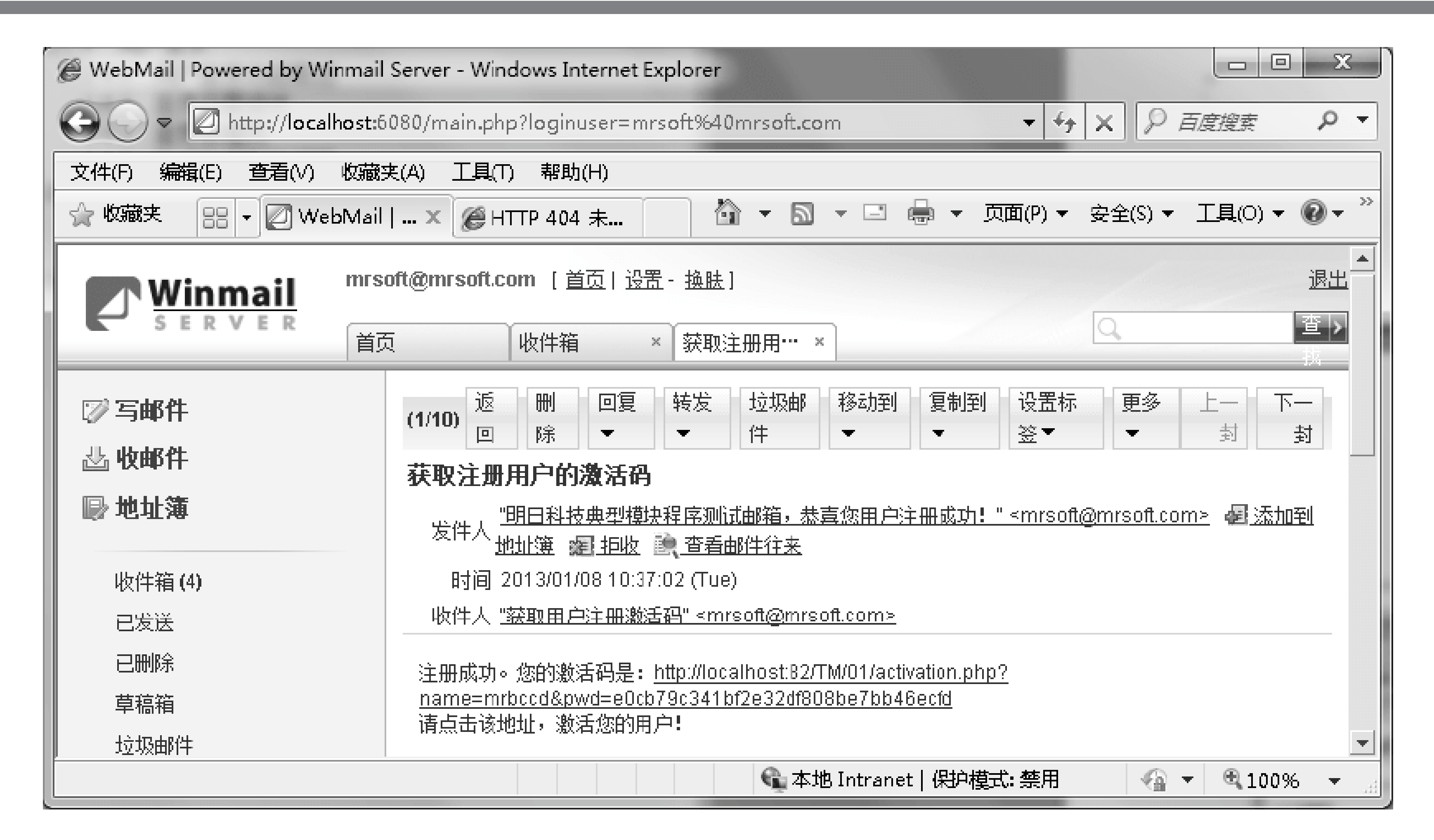


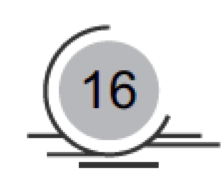
图 1.9　激活邮件的运行结果

当用户进入激活页（activation.php）后，激活页将根据 name 和 pwd 字段来查找匹配的注册用户，如果找到将进行更新操作。激活成功后，将用户名保存到 Session 中，并跳转到相应的页面。代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once("conn/conn.php");
if (!empty($_GET['name']) && !is_null($_GET['name'])){                    //激活注册用户
$num=$conne->getRowsNum("select * from tb_member
where name='".$_GET['name']."' and password = '".$_ GET['pwd']."'");
if ($num>0){
$upnum=$conne->uidRst("update tb_member set active = 1
where name='".$_GET['name']."' and password = '".$_GET['pwd']."'");
if($upnum > 0){
$_SESSION['name'] = $_GET['name'];
echo "<script>alert('用户激活成功！ ');window.location.href='main.php';</script>";
}else{
echo "<script>alert('您已经激活！ ');window.location.href='main.php';</script>";
}
}else{
echo "<script>alert('用户激活失败！ ');window.location.href='register.php';</script>";
}
}
?>
```

1.3.6　免激活用户注册

1.3.5 节介绍的用户注册功能中，需要将激活信息以邮件的形式发送到用户指定的邮箱中进行激活后，用户才可以使用。这里向读者介绍一种不需要激活的用户注册方式，同样以上面注册提交的数据为基础，只需要修改用户注册提交数据的处理方式，就可以实现免激活的用户注册功能。该操作在



register_check.php 文件中完成。关键代码如下：

```
<?php
include_once 'conn/conn.php';                    //包含数据库连接操作文件
$reback = '0';                                   //定义初始变量
$sql = "insert into tb_member(name,password,question,answer,email,realname,birthday,telephone,qq,active)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."','".$_GET['answer']."',
'".$_GET['email']."','".$_GET['realname']."','".$_GET['birthday']."','".$_GET['telephone']."','".$_GET['qq']."','1')";
$num = $conne->uidRst($sql);                     //执行添加语句
if($num == 1){
$reback = '1';
}
echo $reback;                                    //输出变量值
?>
```

说明 在本模块中要使用免激活模式来注册用户，首先应该在 js\register.js 文件中，修改注册数据处理 URL 路径，将注册数据处理文件（register_chk.php）修改为 register_check.php。代码如下：

```
$('regbtn').onclick = function(){
$('imgdiv').style.visibility = 'visible';
url = 'register_chk.php?name='+$('regname').value+'&pwd='+$('regpwd1').value+'&email='+$('email').value;
url += '&question=' +$('question').value+'&answer='+$('answer').value;
url += '&realname=' +$('realname').value+'&birthday='+$('birthday').value;
url += '&telephone='+$('telephone').value+'&qq='+$('qq').value;
xmlhttp.open('get',url,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readyState == 4){
if(xmlhttp.status == 200){
msg = xmlhttp.responseText;
if(msg == '1'){
alert('注册成功，请到您的邮箱中获取激活码！');
location='index.php';
}else if(msg == '-1'){
alert('您的服务器不支持 Zend_mail，或者邮箱填写错误。请仔细检查！！');
}else{
alert(msg);
}
$('imgdiv').style.visibility = 'hidden';
}
}
}
xmlhttp.send(null);
}
```

如此，在执行用户注册数据的处理时，就会调用上面重新编写的 register_check.php 文件，用户注册成功后可以直接登录，而不必再进行激活操作。

1.4 用 户 登 录

1.4.1 用户登录功能概述

用户登录功能就是对用户输入的账号、密码以及验证码进行验证。验证包括信息是否为空、验证码是否错误、用户计算机是否已经被锁定、用户是否存在、用户是否激活、用户名和密码是否匹配、用户登录次数是否达到上限等。登录功能的运行结果如图 1.10 所示。



图 1.10 单击“登录”按钮后的运行结果

登录功能包含 3 个核心文件，分别是 login.php（用户登录页）、login.js（登录信息验证脚本）和 login_chk.php（登录处理页）。

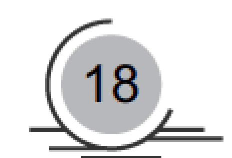
1.4.2 登录页面设计

1．用户登录页（login.php）

相对于注册页面来说，登录页（login.php）中的表单元素要少很多，也没有那么复杂。登录页中的元素名称和属性如表 1.2 所示。

表 1.2 登录页中的元素名称和属性

元 素 名 称	元素属性值	说 明
script	"js/xmlhttprequest.js"	载入 xmlhttprequest 对象脚本
script	"js/login.js"	载入登录信息验证脚本
text	id="lgname" name="lgname"	登录名称
password	id="lgpwd" name="lgpwd"	登录密码





续表

元 素 名 称	元素属性值	说　　明
text	id="lgchk" name="lgchk"	验证码
img	id='chkid' src="" width="55" height="18"	图形验证码
a	id="changea"	更换验证码超链接
hidden	id = "chknm" name = "chknm" value=""	隐藏域，保存验证码
button	id="lgbtn"	“登录”按钮
button	id="reg"	跳转到注册页
button	id="found"	找回密码

2．登录信息验证脚本（login.js）

（1）在登录页 login.php 中，载入了 login.js 脚本文件，通过该脚本文件完成对登录用户的所有验证操作。代码如下：

```
function $(id){
return document.getElementById(id);
}
window.onload = function(){
…
}
```

（2）焦点自动下移。当焦点移动到最后一个文本框即验证码输入框后，再次按下 Enter 键，将调用 chklg()函数对用户输入的信息进行验证。实现原理将在 1.6 节进行详细讲解。代码如下：

```
$('lgname').focus();                              //光标定位到用户名输入框
$('lgname').onkeydown = function(){               //处理用户名输入框的键盘按下事件
   if(event.keyCode == 13){                       //当按下 Enter 键时，光标自动下移
   $('lgpwd').select();                           //密码框获取焦点
}
}
$('lgpwd').onkeydown = function(){
if(event.keyCode == 13){                          //处理密码框的鼠标按下事件
   $('lgchk').select();                           //验证码文本框获得焦点
}
}
$('lgchk').onkeydown = function(){                //处理验证码输入框的鼠标按下事件
if(event.keyCode == 13){
chklg();                                          //调用 chklg()函数
}
}
$('lgbtn').onclick = chklg;
```

（3）进行最基本的检测。包括用户名要合法、信息不能为空、验证码输入正确等。代码如下：

```
function chklg(){
/*判断用户名是否合法，用户名第一位必须是英文字母，不允许有特殊字符*/
```

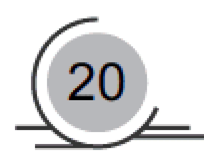
```
if($('lgname').value.match(/^[a-zA-Z_]\w*$/) == null){
alert('请输入合法名称');
$('lgname').select();
return false;
}
/*用户名是否为空*/
if($('lgname').value == ''){
alert('请输入用户名!');
$('lgname').focus();
return false;
}
/*密码是否为空*/
if($('lgpwd').value == ''){
alert('请输入密码!');
$('lgpwd').focus();
return false;
}
/*验证码是否为空*/
if($('lgchk').value == ''){
alert('请输入验证码');
$('lgchk').select();
return false;
}
/*验证码输入是否正确*/
if($('lgchk').value != $('chknm').value){
alert('验证码输入错误');
$('lgchk').select();
return false;
}
```

（4）判断客户端的 Cookie 值是否大于 3，如果大于 3，说明用户是非法操作的，将停止继续执行。代码如下：

```
count = document.cookie.split(';')[0];                    //获取 Cookie 信息
if(count.split('=')[1] >= 3){
alert('因为您的非法操作，您将无法再执行登录操作');
return false;
}
```

（5）将用户输入的信息传递给 login_chk .php 页进行处理。代码如下：

```
url = 'login_chk.php?act='+(Math.random())+'&name='+$('lgname').value+'&pwd='+$('lgpwd').value;
xmlhttp.open('get',url,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4){
if(xmlhttp.status == 200){
//获取 login_chk.php 页的返回值
```



```
msg = xmlhttp.responseText;
if(msg == '0'){
alert('您还没有激活，请先登录邮箱进行激活操作。');
}else if(msg == '1'){
alert('用户名或密码输入错误，您还有 2 次机会');
$('lgpwd').select();
}else if(msg == '2'){
alert('用户名或密码输入错误，您还有 1 次机会');
$('lgpwd').select();
} else if(msg == '4'){
   alert('用户名输入错误');
   $('lgname').select();
} else if(msg == '-1'){
alert('登录成功');
location = 'main.php';
}else{
alert(msg);
}
}
}
}
xmlhttp.send(null);
}
```

1.4.3　登录处理页

在登录处理页（login_chk.php）中，对表单中提交的用户名、密码进行验证，包括用户名是否存在、用户名是否被激活、用户登录次数、密码是否正确等，最终返回验证结果。代码如下：

```
<?php
session_start();                                     //开启 Session 支持
header('Content-Type:text/html;charset=gb2312');     //设置头信息
include_once 'conn/conn.php';                        //载入数据库类
$name = $_GET['name'];
$pwd = $_GET['pwd'];                                 //获取名称和密码
if(!empty($name) and !empty($pwd)){
//得到当前数据表中的 count 和 active 字段
$sql = "select name,count,active from tb_member where name = '".$name."'";
$active = $conne->getFields($sql,2);
$count = $conne->getFields($sql,1);
$conne->close_rst();
//判断用户是否存在
if($active == ''){
if(is_null($_COOKIE['count']) or $_COOKIE['count'] == 0){
setcookie('count',1);
}else{
setcookie('count',$_COOKIE['count']+1);
```

```
}
$reback = 4;
}else if($active == 0){                          //判断用户是否被激活
$reback = '0';
}else if($count >= 3){
$reback = '3';                                   //判断用户的登录次数
}else{
$sql .= " and password = '".md5($pwd)."'";
$num = $conne->getRowsNum($sql);
if($num == 0 or $num == ''){                     //如果用户密码错误，登录次数加 1
$num = $conne->uidRst("update tb_member set count = ".($count+1)." where name = '".$name."'");
$reback = ($count+1);
}else{                                           //登录成功，清空 count 字段值
if($count != 0){
$num = $conne->uidRst("update tb_member set count = 0 where name = '".$name."'");
}
//设置 Cookie
if(isset($_COOKIE['count']) and $_COOKIE['count'] != 0){
setcookie('count',0);
}
$_SESSION['name'] = $name;
$reback = '-1';
}
}
}
echo $reback;
?>
```

1.4.4 生成及刷新验证码

验证码生成应用的是 GD2 函数库中的函数，操作存储于 valcode.php 文件中，其完整代码可以参考 1.6 节，这里不再赘述。

当页面第一次被载入时，将执行 login.js 中的 showval()函数来生成验证码，并且调用 valcode.php 文件以图像形式输出验证码。因为验证码中带有干扰串，如果用户看不清，可以单击首页中的“看不清”超链接来再次执行 showval()函数，即刷新验证码。login.js 脚本中 showval()函数的代码如下：

```
showval();                                       //调用 showval()函数生成图文验证码
$('changea').onclick = showval;                  //更换验证码
//验证码生成函数
function showval(){
num = '';
for(i=0;i<4;i++){
   …                                             //生成随机数
}
$('chkid').src='valcode.php?num='+num;           //显示图片验证码
$('chknm').value = num;                          //将验证码保存到 chknm 隐藏域
}
```



1.5　找 回 密 码

1.5.1　找回密码功能概述

找回密码功能用到的技术也是邮箱技术。用户输入正确的账号、密保问题、密保答案后，系统将随机生成的一组数字发送到用户的邮箱中，用户可以通过邮箱来得到新密码。找回密码页面的运行结果如图 1.11 所示。



图 1.11　找回密码页面的运行结果

1.5.2　找回密码功能实现过程

找回密码操作从单击主页中的“找回密码”按钮开始，在弹出的 found.php 文件中创建一个表单，提交要找回的账号、密保问题和密保答案，并且载入 xmlhttprequest.js 和 found.js 脚本文件。found.php 的关键代码如下：

```
<link rel="stylesheet" href="css/style.css" />
<script language="javascript" src="js/found.js"></script>
<script language="javascript" src="js/xmlhttprequest.js"></script>
<div id="fdbgdiv" >
<div id="top"> >>密码找回</div>
<div id="foundnamediv">找回账号：
<input id="foundname" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
<div id="foundnamediv">密保问题：
<input id="fdquestion" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
<div id="foundnamediv">密保答案：
<input id="fdanswer" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></div>
<div id="foundnamediv" align="center"><button id="step1"></button></div>
</div>
```

在 found.js 文件中，调用 found_chk.php 文件，完成密码的找回操作。关键代码如下：

```
function $(id){
return document.getElementById(id);
}
window.onload = function(){
$('foundname').focus();
$('step1').onclick = function(){
if($('foundname').value != '' && $('fdquestion').value != '' && $('fdanswer').value != ''){
xmlhttp.open('get','found_chk.php?foundname='+$('foundname').
value+'&question='+$('fdquestion').value+'&answer='+$('fdanswer').value,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readyState == 4 && xmlhttp.status == 200){
```

```
msg = xmlhttp.responseText;
if(msg == '1'){
alert('找回密码成功，请登录邮箱取回密码!');
window.close();
}else{
alert('填写信息错误！');
}
}
}
xmlhttp.send(null);
}else{
alert('请填写完整信息');
$('foundname').focus();
return false;
}
}
}
```

在 found_chk.php 文件中，对用户提交的密保问题和密保答案进行验证，如果正确则生成新的密码，并且更新数据库中存储的密码，同时将新密码发送到用户注册时填写的邮箱中。found_chk.php 的关键代码如下：

```
<?php
include_once 'conn/conn.php';
require_once 'Zend/Mail.php';                       //调用发送邮件的文件
require_once 'Zend/Mail/Transport/Smtp.php';        //调用 SMTP 验证文件
$reback = '0';
$name = $_GET['foundname'];                         //获取用户名
$question = $_GET['question'];                      //获取密码保护问题
$answer = $_GET['answer'];                          //获取密码保护的答案
$sql = "select email from tb_member
where name = '".$name."' and question = '".$question."' and answer = '".$answer."'"; //定义查询语句
$email = $conne->getFields($sql,0);                 //判断用户提交的密保问题和答案是否正确
if($email != ''){                                   //如果正确则执行如下操作
$rnd = rand(1000,time());                           //生成随机数
$sql = "update tb_member set password = '".md5($rnd)."'
where name = '".$name."' and question = '".$question."' and answer = '".$answer."'"; //更新密码
$tmpnum = $conne->uidRst($sql);                     //执行更新操作
if($tmpnum >= 1){
//发送密码邮件
$subject="找回密码";
$mailbody='密码找回成功。您账号的新密码是'.$rnd;
//$envelope["from"]="cym3100@163.com";
$envelope="mrsoft@mrsoft.com";                      //网络版定义登录使用的邮箱
/*SMTP 测试版发送邮件方式，使用 SMTP 作为服务器*/
$tr = new Zend_Mail_Transport_Smtp('192.168.1.247');
$mail = new Zend_Mail();
$mail->addTo($email,'获取用户新密码');
$mail->setFrom($envelope,'明日科技典型模块程序测试邮箱，修改用户注册密码！');
```

```
$mail->setSubject($subject);
$mail->setBodyHtml($mailbody);
//$mail->send($tr);
/*网络版发送邮件方法*/
/*$config = array('auth' => 'login',
'username' => 'mrsoft@mrsoft.com',
'password' => ' mingri');                    //定义 SMTP 的验证参数
$transport = new Zend_Mail_Transport_Smtp('smtp.sohu.com', $config);        //实例化验证的对象
$mail = new Zend_Mail('GBK');                //实例化发送邮件对象
$mail->setBodyHtml($mailbody);               //发送邮件主体
$mail->setFrom($envelope, '明日科技典型模块程序测试邮箱，修改用户注册密码！');
$mail->addTo($email, '获取用户新密码');        //定义邮件的接收邮箱
$mail->setSubject($subject);                 //定义邮件主题
$mail->send($transport);                     //执行发送操作*/
/*网络版发送邮件方法*/
if(false ==$mail->send($tr) ){
$reback = '-1';
}else{
$reback = '1';
}
}else{
$reback = '2';
}
}else{
$reback = $sql;
}
echo $reback;
?>
```

1.6　技术提炼

1.6.1　防 SQL 注入技术

SQL 注入，是指一些精通 SQL 语句的用户，通过在表单或浏览器地址栏中输入 SQL 语句来绕过系统检验的一种技术。一般防范 SQL 注入的手段是过滤敏感字符，例如引号等。

PHP 相对于 ASP 来说，要安全得多，但这不代表 PHP 网站就不会出现 SQL 注入的情况。幸运的是，在 PHP 中防 SQL 注入，要比 ASP 简单、方便，不需要写一大段的转换语句，只要几个函数即可。

（1）intval()函数

intval()函数的作用是返回变量的整数值。其语法如下：

```
int intval( mixed var [, int base] )
```

（2）addslashes()函数

addslashes()函数就是在操作数据库时，对其中的特殊字符进行自动转义，即在特殊字符前加上反

斜杠（\），包括单引号（'）、双引号（"）、NULL，但是不包括“%”和“_”。该函数的语法如下：

```
string addslashes( string str )
```

此外，还可以使用 mysql_real_escape_string()来进行转义，效果和 addslashes()函数是一样的。

1.6.2 Ajax 无刷新验证技术

Ajax（Asynchronous JavaScript and XML，异步 JavaScript 和 XML）是时下最流行的技术。Ajax 不是新的技术，而是原有技术的集合，这从它的名字上就能够看出来。

Ajax 的核心技术是 xmlHttpRequest。通过 xmlHttpRequest 中的 open()和 send()方法，可以在不刷新当前页面的情况下向处理页发送数据；通过 xmlHttpRequest 中的 responseText 和 responseXML 属性，可以得到处理页的输出结果。

Ajax 之所以能够流行，是因为它能为用户提供良好的交互性，这个特性在本模块中被充分地体现出来。注册用户无须经过“漫长”的等待就可以知道输入的账号是否可用，登录用户也不必担心因为输错登录信息而重新刷新整个页面。

使用 Ajax，一般分为下面几步。

（1）创建 xmlHttpRequest 对象。不同的浏览器，创建 xmlHttpRequest 对象及使用的方法有一些差别，这里只针对 IE 浏览器进行创建。代码如下：

```
var xmlhttp = false;                                    //初始化变量
//如果 ActiveXObject 存在，说明是 IE 1.0 以上的版本，否则使用 XMLHttpRequest 创建
if(window.ActiveXObject){
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}else if(window.XMLHttpReuqest){
xmlhttp = new XMLHttpRequest();
}
```

（2）对象创建成功后，就可以使用对象中的 open()方法创建新请求了。该方法的格式如下：

```
xmlhttp.open(rmethod,rurl,isAsync);
```

☑ rmethod 参数指明请求的方法，如 get 或 post。
☑ rurl 参数指明请求的页面。可以是绝对地址，也可以是相对地址。
☑ isAsync 参数指明请求是否为异步。默认为 true，即异步。

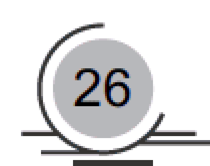
（3）如果 isAsync 等于 true，那么当请求的状态改变时，将调用 onreadystatechange 属性，该属性指定了一个回调函数。格式如下：

```
xmlhttp.onreadystatechange = reabackfunc;
```

或者

```
xmlhttp.onreadystatechange = function(){…}
```

（4）在回调函数中，首先需要判断 http 的请求状态和 http 状态码。这是通过 readystate 和 statues





属性来判断的。readystate 属性有 5 种状态值，常用的是 4，表示数据接收完毕。statues 属性的值比较多，常用的是 200，表示请求成功。一般使用这两个属性一起来判断。格式如下：

```
xmlhttp.onreadystatechange = function(){
        if(readystate == 4 and statues == 200){
            ...
        }
}
```

（5）当响应页处理结束后，即满足了“readystate==4 and statues == 200”这个条件，就可以使用 xmlHttpRequest 对象的属性获取响应页的值了。常用的有 responseText、responseXML、responseStream 等，这里以 responseText 为例进行介绍。

responseText 属性是将响应页的输出信息作为字符串返回。格式如下：

```
str = xmlhttp.responseText;
```

（6）使用 send()方法来接收回应。send()方法可以传递数据，但这取决于 open()方法中的 rmethod 参数，当参数为 get 时，数据是附在 URL 中进行传递的；当参数为 post 时，数据只能使用 send()方法进行传递。send()方法的语法如下：

```
xmlhttp.send([rdate]);
```

1.6.3　验证码技术

验证码技术，是为了防止用户名和密码被暴力破解，在登录页面上生成一组随机数，每次刷新页面时，随机数都会改变。一般是 4 位，也有更多位的。

1．获取验证码

PHP 中的验证码可以通过 rand()函数生成随机数的方式得到。rand()函数可以获取指定范围内的随机数。该函数的语法如下：

```
int rand([int min, int max])
```

如果省略两个参数，那么将返回 0 到 RAND_MAX 之间的随机整数；否则，返回 min 和 max 之间的整数。例如本例中要获取 4 位十六进制的整数，代码如下：

```
<?php
for($i=0;$i<4;$i++){
$num .= dechex(rand(0,15));                    //生成随机数
}
...
?>
```

函数 dechex()可以将参数转换为十六进制表示。

使用 JavaScript 也可以生成十六进制随机数，但是稍有些复杂。JavaScript 中不能直接将十进制数



转为十六进制，需要手动进行转换。首先使用 Math.random()函数生成 0～15 之间的随机数，然后使用 Math.ceil()函数将随机数取整，接下来就要逐次判断该值，如果该值大于 9，那么将 10～15 的数一一对应转换为 a、b、c、d、e、f。转换完成后，将值累加，最后传给 valcode.php 页。

使用 JavaScript 生成十六进制随机数的完整代码如下：

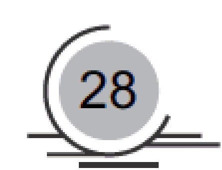
```
//生成随机数
function showval(){
num = '';
for(i=0;i<4;i++){                                  //循环输出 4 位验证码
tmp =  Math.ceil((Math.random() * 15));            //取得一位十六进制的整数
if(tmp > 9){                                       //依次判断随机数
switch(tmp){
case(10):                                          //如果随机数等于 10，换为 a
num += 'a';
break;
case(11):
num += 'b';                                        //如果随机数等于 11，换为 b
break;
case(12):
num += 'c';                                        //如果随机数等于 12，换为 c
break;
case(13):                                          //如果随机数等于 13，换为 d
num += 'd';
break;
case(14):                                          //如果随机数等于 14，换为 e
num += 'e';
break;
case(15):                                          //如果随机数等于 15，换为 f
num += 'f';
break;
}
}else{
num += tmp;
}
}
$('chkid').src='valcode.php?num='+num;             //将生成的随机数传给图像生成页
$('chknm').value = num;                            //将随机数的值保存到页面的隐藏域中
}
```

2. 显示随机数图片

显示随机数的方式很多，将随机数写入一个图片中再显示是目前常用的方法。在 PHP 中，可以使用 GD 函数库来实现。使用到的函数主要有 imagecreate()、imagecolorallocate()、imagestring()、imagesetpixel()、imagepng()和 imagedestroy()函数。

（1）imagecreate()函数

imagecreate()函数用来创建一个基于调色板的空白图像源，这是生成图片的第一步。该函数的语法如下：



```
resource imagecreate( int width, int height )
```

参数 width 和 height 分别指定了图像的宽和高。

（2）imagecolorallocate()函数

imagecolorallocate()函数可以为创建后的图像分配颜色。该函数的语法如下：

```
int imagecolorallocate( resource image, int red, int green, int blue )
```

☑ 参数 image 是一个图像源。
☑ 参数 red、green 和 blue 表示红、黄、蓝三元素的成分。每种颜色的取值范围在 1～255 之间。

（3）imagestring()函数

图像创建完成后，就可以使用 imagestring()函数来添加图像文字了。该函数的语法如下：

```
bool imagestring( resource image, int font, int x, int y, string s, int col )
```

☑ 参数 image 是一个图像源。
☑ 参数 font 可以设置字体，如果使用系统默认字体，可以使用 1～5 的数字。
☑ 参数 x 和 y 分别表示文字相对于整幅图像的 x 轴和 y 轴坐标，即所输入的字符串的左上角坐标。
☑ 参数 s 是要显示的字符串。
☑ 参数 col 为字体颜色，也是使用 imagecolorallocate()函数来分配。

（4）imagesetpixel()函数

使用 imagecolorallocate()函数创建的是一个单一背景色的图像，如果希望向图像中添加干扰码，可以使用 imagesetpixel()函数，该函数的作用是画一个像素点。imagesetpixel()函数的语法如下：

```
bool imagesetpixel( resource image, int x, int y, int color )
```

参数说明和 imagestring()函数相似，这里不再赘述。

（5）imagepng()函数

imagepng()函数将创建完成的图片以.png 的格式输出。该函数的语法如下：

```
bool imagepng( resource image [, string filename] )
```

☑ 参数 image 是要保存的图像源。
☑ 参数 filename 是要保存的图像名。如果省略，则直接输出到浏览器。

（6）imagedestroy()函数

图像保存完毕后，使用 imagedestroy()函数来释放内存。该函数的语法如下：

```
bool imagedestroy( resource image )
```

验证码的生成文件是 valcode.php。其完整代码如下：

```
<?php
header("content-type:image/png");                               //设置页面编码
$num = $_GET['num'];                                            //获取超级链接传递的随机数
```

```
<? gewidth=60;                                                    //定义画布的宽
$imageheight=18;                                                  //定义画布的高
$numimage = imagecreate($imagewidth,$imageheight);                //创建画布
imagecolorallocate($numimage,240,240,240);                        //设置画布颜色
for($i=0;$i<strlen($num);$i++){                                   //循环读取随机数
$x = mt_rand(1,8)+$imagewidth*$i/4;
$y = mt_rand(1,$imageheight/4);
$color=imagecolorallocate($numimage,mt_rand(0,150),mt_rand(0,150),mt_rand(0,150)); //定义图像的颜色
imagestring($numimage,5,$x,$y,$num[$i],$color);                   //将随机数写入到画布中
}
for($i=0;$i<200;$i++){                                            //for 循环语句生成干扰线
$randcolor=imagecolorallocate($numimage,rand(200,255),rand(200,255),rand(200,255)); //定义颜色
imagesetpixel($numimage,rand()%70,rand()%20,$randcolor);          //生成干扰线
}
imagepng($numimage);                                              //生成图像
imagedestroy($numimage);                                          //释放资源
?>
```

1.6.4　E-mail 激活技术

为了防止恶意注册，现在有很多网站采用了 E-mail 激活技术。当用户注册成功后，并不是马上可以使用，而是需要先登录邮箱，通过系统发出的 E-mail 中的链接进行激活，只有激活后，账号才可以使用。对于找回密码的用户，系统会将新密码发送到邮箱中，这在一定程度上提高了安全性。

本模块使用 ZendFramework 中的 Zend_Mail 组件完成邮件的发送操作，在本书的光盘中没有提供 Zend_Mail 组件，如果要运行这个程序，需要读者自己下载 Zend 组件，并将其复制到本模块的根目录下，即 mr/01 文件夹下。

> **说明** 在执行邮件的发送时，笔者使用的是 Winmail 软件，邮箱名称是 mrsoft@mrsoft.com，密码是 mingri。读者在完成用户注册之后，可以登录这个邮箱完成注册用户的激活操作，否则注册的用户是不能够使用的。

注册用户激活的原理是控制数据库中用户信息表（tb_member）中 active 字段的值，注册成功后该字段的默认值是 0，但此时该用户是不能够登录网站的，必须通过邮箱激活，将该字段的值更新为 1 之后才可以使用。激活操作的文件存储于 activation.php 文件中。关键代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once("conn/conn.php");
if (!empty($_GET['name']) && !is_null($_GET['name'])){                //激活注册用户
$num=$conne->getRowsNum
 ("select * from tb_member where name='".$_GET['name']."' and password = '".$_ GET['pwd']."'");
if ($num>0){
$upnum=$conne->uidRst
 ("update tb_member set active = 1 where name='".$_GET['name']."' and password = '".$_GET['pwd']."'");
```

```
if($upnum > 0){
  $_SESSION['name'] = $_GET['name'];
echo "<script>alert('用户激活成功！');window.location.href='main.php';</script>";
}else{
  echo "<script>alert('您已经激活！');window.location.href='main.php';</script>";
}
}else{
echo "<script>alert('用户激活失败！');window.location.href='register.php';</script>";
}
}
?>
```

1.6.5　应用键盘响应事件验证信息是否合法

新用户注册时，随着信息输入，系统实时显示信息的正确性；用户登录时，无须使用鼠标来选择文本框，只要按下 Enter 键，焦点就自动下移。这些功能都是通过 JavaScript 脚本中的键盘响应事件来实现的。随着 Ajax 技术的流行，JavaScript 已经成为一个开发人员必须要掌握的技术。下面就来简单了解一下键盘事件。

用户通过 onkeydown 和 onkeyup 事件来触发响应事件。使用方法和 onclick 事件类似。onkeydown 表示当键盘上的键被按下时触发，onkeyup 则正好相反，当键盘上的键被按下又抬起时触发。在页面中加载事件的方式有多种，这里介绍两种最常用的方式。

（1）将事件直接添加到页面元素中。这种方法最直接、简单。格式如下：

```
<script type="text/javascript">
      …
      function refer(){
         …
      }
</script>
<input type="text" onkeydown="refer()" />
```

当该用户输入完信息后，按任意键，onkeydown 事件被触发，并调用 refer()函数。

（2）通过 window.onload 加载。当页面被载入时，事件被载入。格式如下：

```
<script>
      window.onload = function(){
          document.getElementById('lgname').onkeydown = function(){
               …
          }`
      }
</script>
…
<input id="lgname" type="text" />
…
```

当用户输入信息时，每输入一个字母，都将触发该事件，在该事件调用的函数中，对用户输入信

息进行判断，例如用户名必须大于等于 2、密码最短 6 位、E-mail 必须合法等。验证用户名的代码如下：

```
//验证用户名
//为 id 等于 regname 的页面元素添加 onkeyup 事件
//用户每按一次键，都会调用一次该函数
$('regname').onkeyup = function (){
name = $('regname').value;                          //获取输入内容
cname2 = '';
if(name.match(/^[a-zA-Z_]*/) == ''){                //判断输入字符是否在有效范围之内
$('namediv').innerHTML = '<font color=red>必须以字母或下划线开头</font>';
cname1 = '';
}else if(name.length < 2){                          //判断输入字符的长度
$('namediv').innerHTML = '<font color=red>注册名称必须大于等于 2 位</font>';
cname1 = '';
}else{
$('namediv').innerHTML = '<font color=green>注册名称符合标准</font>';
cname1 = 'yes';
}
chkreg();
}
…
```

该段代码的运行结果如图 1.12 所示。

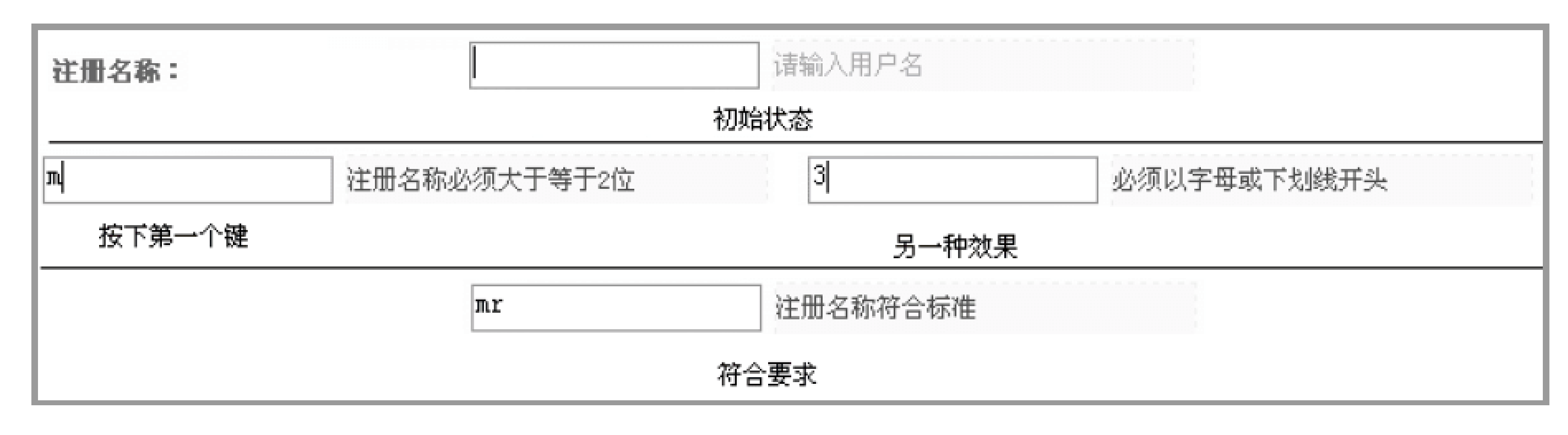
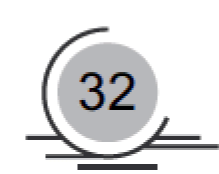


图 1.12　验证信息合法性

使用 onkeydown 事件还可以实现对特定键的控制，包括 Enter 键、Ctrl 键、Alt 键等在内的所有按键，这是通过在 onkeydown 事件中使用 keyCode 属性来实现的。keyCode 属性能够知道用户按下的是哪个键，例如 Enter 键等于 13、空格键等于 32 等。使用 keyCode 属性的一般格式如下：

```
<script>
    window.onload = function(){
        document.getElementById('lgname') = function(){
            if(event.keyCode == 13){                    //判断用户是否按下了 Enter 键
                …
            }
        }`
    }
</script>
```

在本模块中，实现了焦点自动下移功能。当用户按下 Enter 键时，页面的焦点自动移到下一个文本框中。实现后的效果如图 1.13 所示。





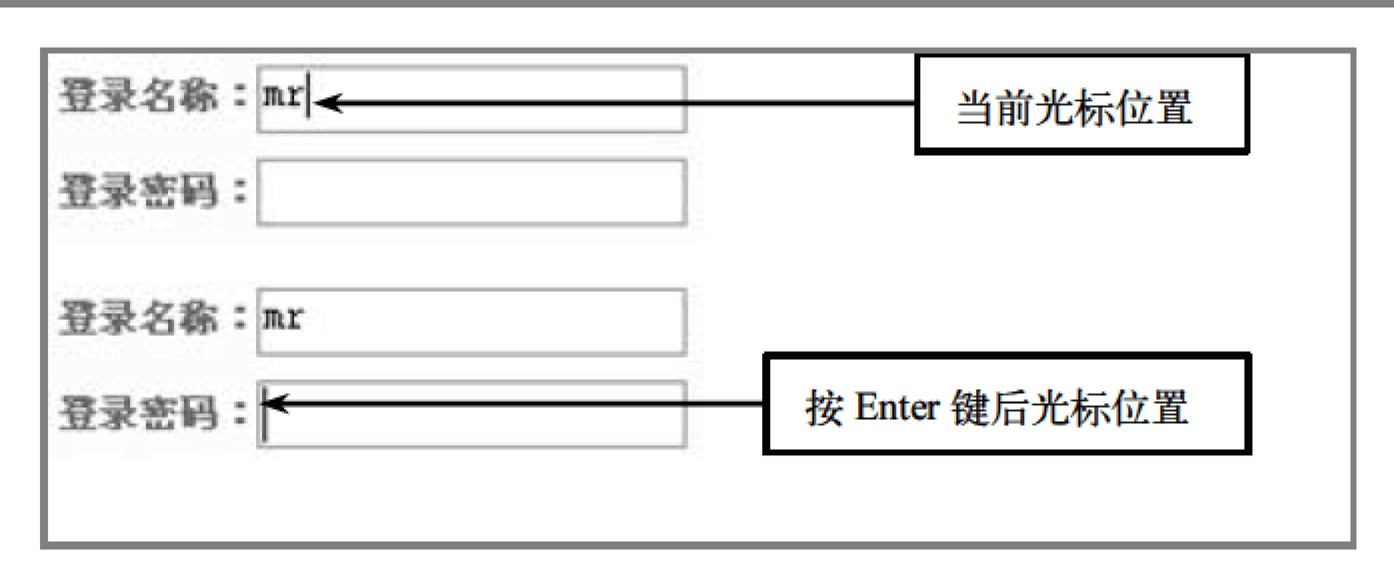


图 1.13　焦点自动下移的运行效果

1.6.6　PHP 中操作 Cookie 技术

Cookie 的作用是当用户第一次访问某服务器时，服务器将一些信息保存到客户端计算机中。在以后的一段时间内，当用户再次访问这个服务器时，服务器通过 Cookie 信息，能够识别该用户。在很长一段时间内，Cookie 都被当作一个不安全的因素，很少再被使用，转而使用 Session。但是，Cookie 的功能确实十分方便，如自动登录、网站统计等，都要比其他的实现方式要简单、快捷得多。Cookie 只是一个文本文件，不能访问本地硬盘，无法传播病毒木马程序。唯一要注意的是，Cookie 只能识别计算机，而不会在意谁在使用。

在 PHP 中操作 Cookie 使用 setcookie()函数和$_COOKIE 预定义变量。setcookie()函数的语法格式如下：

```
bool setcookie( string name [, string value [, int expire [, string path [, string domain [, bool secure]]]]] )
```

- ☑ 参数 name 设置了 Cookie 的名字。
- ☑ 参数 value 是 Cookie 的值，即参数 name 的值。
- ☑ 参数 expire 用来设置 Cookie 的过期时间。该参数以时间戳的形式存在。一般设置 Cookie 过期时间时，通过“time()+秒数”来实现。例如“time()+60×10”，表示 Cookie 将在 10 分钟后失效。设置 path 设马上失效，可以将时间设为当前日期之前，如“time()−1”，那么 Cookie 会立即失效。
- ☑ 参数 path 表示 Cookie 在服务器端的有效范围。如果 path 设为“/”，那么 Cookie 在整个服务器内都有效，如果 path 设为“/05/”，那么 Cookie 只在服务器下的 05 目录有效。
- ☑ 参数 domain 指定了 Cookie 有效的域名范围。以 www.mrbccd.com 为例，如果 domain 设为“.mrbccd.com”，那么 Cookie 在该域的所有子域都有效；如果 domain 设为“www.mrbccd.com”，那么 Cookie 只在 www.mrbccd.com 域内有效。
- ☑ 参数 secure 表明是否在 https 中传送。如果是 true，Cookie 仅在 https 连接中被设置，默认是 false。

1.6.7　在 JavaScript 中操作 Cookie 技术

在 JavaScript 中可以通过 document 对象中的 Cookie 属性对 Cookie 进行创建、读取、删除等操作。先来看一下创建 Cookie 的格式（代码中加粗的部分是需要用户设置的）。

```
documen.cookie = "Name=Value; expires=time; path=URL; domain= Domain"
```

其中各个参数的含义和 PHP 中的基本相同，这里不再赘述。

在 JavaScript 中读取 Cookie 有一点复杂。在 JavaScript 看来，Cookie 就是一个字符串。例如一个名字为 count、值为 10 的 Cookie，在 JavaScript 中的格式是：

```
count=10
```

如果有多个 Cookie，那么它的格式如下：

```
count=10; name=mr; …
```

而真正需要的值，其实只有 10 而已。所以，要获取 Cookie 的值，就要使用 split()函数将 Cookie 进行拆分。获取 count 值的代码如下：

```
<script>
if(document.cookie==''){                                    //Cookie 等于空时
document.write('no cookie');
}else{
cookiearr = document.cookie.split('; ');                    //以分号为分隔符，对 Cookie 进行分割
leng = cookiearr.length;                                    //获取数组长度
for(i=0; i<leng; i++){
if(cookiearr[i].split('=')[0] == 'count'){                  //判断哪一个 Cookie 等于 count
document.write(cookiearr[i].split('=')[1]);                 //输出 count 的值
}
}
}
</script>
```

1.6.8 用户自动登录技术

自动登录的原理是：当用户打开登录页面时，登录页面首先判断用户客户端机器中的 Cookie 值，如果该值不存在，或者 Cookie 已失效，将显示登录表单；如果 Cookie 值存在，则直接使用 Cookie 中所保存的用户名进行登录。

在本模块中，用户首先打开的是 index.php 页，该页判断 Cookie 值，如果没有，则跳转到 login.php 进行登录；如果 Cookie 有值，则将 Cookie 中的用户名保存到 Session，直接进入主页面。实现代码如下：

```
<?php
session_start();                                                      //开启 Session 支持
header('Content-Type:text/html;charset=gb2312');                      //设置页面编码
if(!empty($_COOKIE['name']) and !is_null($_COOKIE['name'])){          //判断 Cookie 是否为空
$_SESSION['name'] = $_COOKIE['name'];                                 //将 Cookie 保存到 Session 中
header('location:http://localhost/model/05/01/main.php');             //跳转到 main.php 页
}else{                                                                //Cookie 为空，说明没有登录
header('location:http://localhost:82/TM/01/login.php');               //跳转到 login.php 页
}
?>
```



这里$_COOKIE['name']的值是在登录成功时设置的。相关代码如下：

```
<?php
…                                                           //如果登录成功
setcookie('name',$name,time()+60*10);                      //保存$name 的同时设置时间
…
?>
```

1.7　本章小结

本章主要讲解的是注册登录模块设计过程，即从数据库设计到具体模块的创建。其中在具体模块创建的过程中针对用户登录模块、用户注册模块进行重点讲解，并且在其中应用了很多新的技术和技巧，例如通过 Ajax 无刷新验证技术、通过在 PHP 中操作 Cookie 技术、通过在 JavaScript 中操作 Cookie 技术，以及通过 Zend_mail 函数实现 PHP 操作技术等。

第 2 章

网页计数器模块

（Apache+PHP+MySQL 5.0 实现）

网页计数器的形式多种多样，有直接以数字计数存储于文本或数据库中的，有以图形方式计数存储于文档或数据库中的。在浏览网页时经常可以看到网站的首页带有一个制作精美的图形计数器，用来显示该网站的访问量。应用图形做计数器，不仅可以美化网站的页面，而且还能够吸引更多浏览者的注意力，提高网站的访问量。

通过阅读本章，可以学习到：

- 简单数字计数器
- Session 实现无刷新计数器
- GD2 图形计数器
- 数据库数字计数器
- Cookie 计数器
- 图形数字计数器



2.1　网页计数器模块概述

网站的计数器对于网站管理者来说是一个非常值得关注的部分，它记录了网站被访问的次数，客观地反映了网站受欢迎的程度。如果网站的点击率很高，访问者也会认为网站值得关注，因而耐心驻足，进而增加网站的访问量，提高网站的知名度。

网页计数器模块主要从计数器功能实现的方法着手，详细地介绍了简单的数字计数器、图形数字计数器、GD2 图形数字计数器、数据库数字计数器的实现方法。

网页计数器的操作流程如图 2.1 所示。

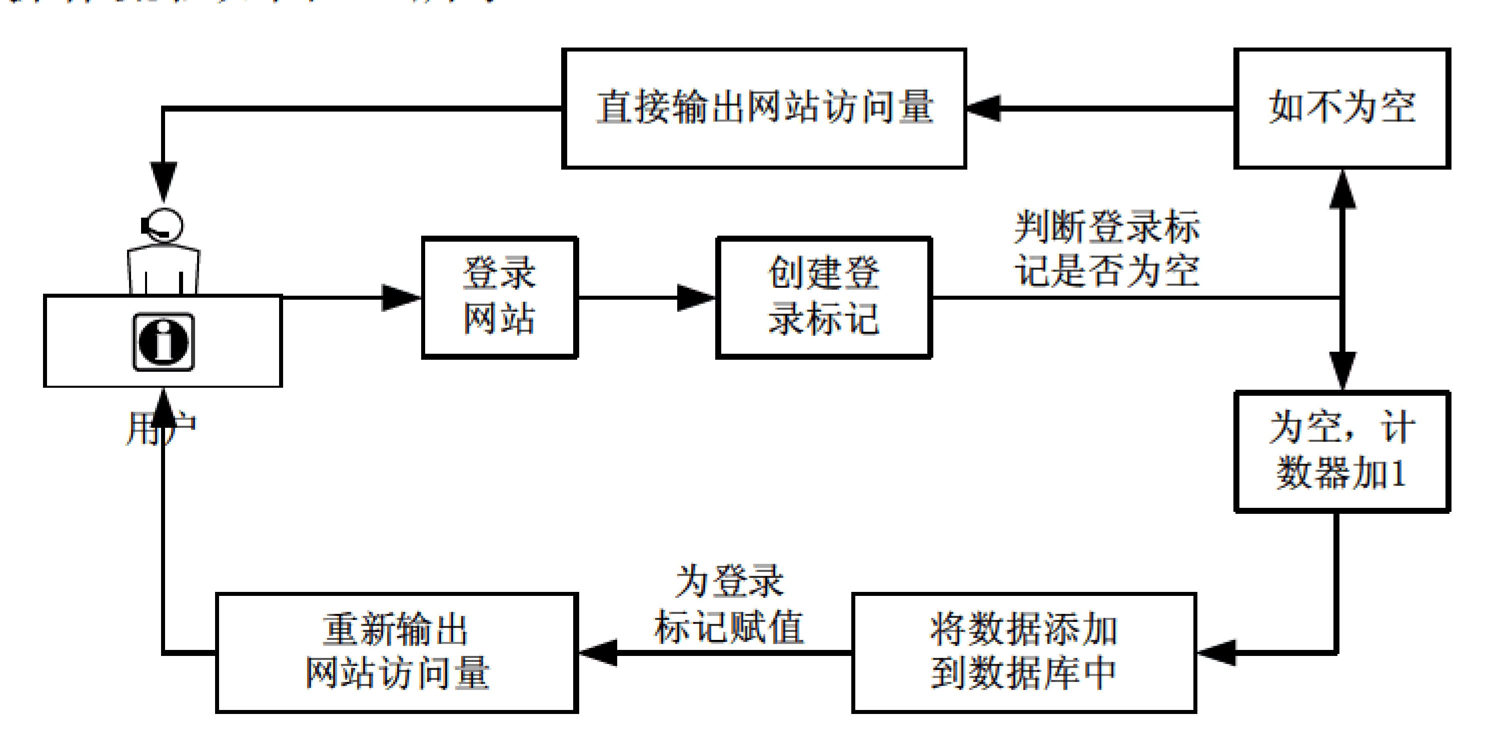


图 2.1　网页计数器的操作流程

2.2　简单数字计数器

数字计数器是最简单的一种，它将数据存储于文本文件中。其思路是首先判断文本文件是否存在，如不存在，则打开失败；如打开成功，则继续执行并读取文件中的数据，将计数器增加 1。然后以写的方式重新打开文件，把新的统计数据写入到文件中，关闭文件。最后重新打开文件，读取并输出文件中的数据。

简单数字计数器的操作流程如图 2.2 所示。

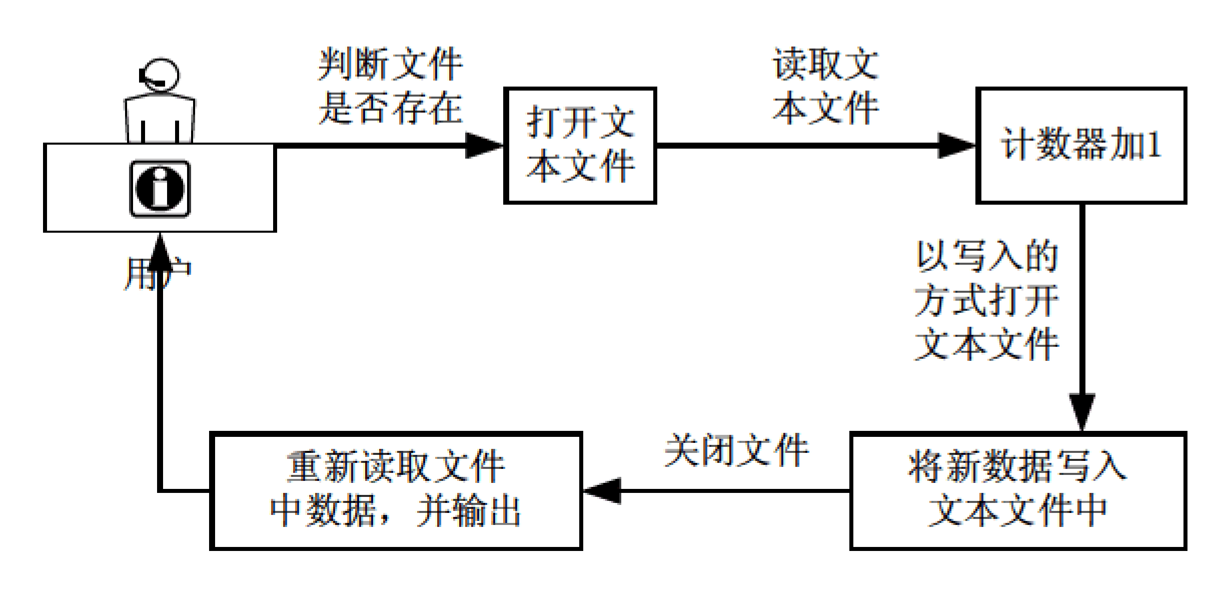


图 2.2　简单数字计数器的操作流程

按上述原理设计的计数器在页面刷新时计数器的值就会增加，所以要想让页面刷新时计数器的数据不增加，最好是应用 Session 变量防止重复计数。

当应用 Session 防止重复计数后，无论如何刷新页面计数器的值都不会增加。只有在重新打开页面时，计数器的值才会继续增加。简单数字计数器的结果如图 2.3 所示。

公司理念：以高新技术为依托，战略性地开发具有巨大市场潜力的高价值的产品。

公司远景：成为拥有核心技术和核心产品的高科技公司，在某些领域具有领先的市场地位。

核心价值观：永葆创业激情、每一天都在进步、容忍失败，鼓励创新、充分信任、平等交流。

网站当前访问量：100

© CopyRights reserved 2008 吉林省明日科技有限公司
客户服务邮箱：mingrisoft@mingrisoft.com

图 2.3　简单数字计数器的结果

分析简单的数字计数器，首先通过 session_start()函数初始化一个 Session 变量。通过 Session 变量来控制页面刷新对计数器的影响。

然后通过 if 语句判断 Session 变量的值是否为空，如果为空，则打开指定的文本文件，否则将不执行。

如果 Session 变量的值为空，则应用 fopen()函数以只读的方式打开指定的文本文件，如果文件不存在，则输出打开文件失败，否则将继续执行读取文件中的数据。这里将指定的文本文件 counter.txt 存储于根目录下。如果文本文件存在，则继续执行，将应用 fgets()函数读取文本文件中的数据，并通过 fclose()函数关闭文件。此时将读取到的数据$counter 加 1，即将计数器中的值加 1。通过 fopen()函数重新打开指定的文本文件，注意是以写入的方式打开，并应用 fputs()函数将新统计的数据写入到文本文件中，关闭文件。

最后为 Session 变量赋值，将 Session 变量赋值为 1，用于控制因网页刷新而导致计数器的值无限增加。

简单计数器的文件存储于 TM\02\01\index.php 文件中。关键代码如下：

```
<?php
//使用文本存储数据
if(!isset($_SESSION['temp'])){   //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义变量
if(($fp=fopen("counter.txt","r"))==false){
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);                      //读取文件中的数据
fclose($fp);                                   //关闭文本文件
$counter++;                                    //计数器增加 1
$fp=fopen("counter.txt","w");                  //以写的方式打开文本文件
fputs($fp,$counter);                           //将新的统计数据增加 1
fclose($fp);
}                                              //关闭文件
$_SESSION['temp']=1;                           //计数器的值增加后，为$_SESSION[temp]赋值 1
}
//从文本文件中读取统计数据
if(($fp=fopen("counter.txt","r"))==false){
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);
```



```
fclose($fp);
}
?>
```

（1）fopen()函数用于打开某文件，并返回该文件的标识指针。该函数的语法如下：

```
resource fopen(string filename, string mode [, int use_include_path [, resource context]])
```

fopen()函数的参数说明如表 2.1 所示。

表 2.1　fopen()函数的参数说明

参　数	说　明
filename	必要参数。用于指定要打开文件的本地或远程地址
mode	必要参数。用于指定要打开文件的模式
use_include_path	可选参数。如果将该参数设置为 true，PHP 会尝试按照 include_path 标准包含路径中的每个指向去打开文件
context	可选参数。设置提高文件流性能的一些选项

在 fopen()函数的 mode 参数中，其指定的打开模式很多，其中常用的有如下几种：

☑　r：只读方式打开，将文件指针指向文件头。

☑　r+：读写方式打开，将文件指针指向文件头。

☑　w：写入方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在则尝试创建。

☑　w+：读写方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在则尝试创建。

（2）fgets()函数用于获取文件指针所在行指定长度的内容。该函数的语法如下：

```
string fgets(int handle [,int length])
```

☑　handle：必要参数。指应用 fopen()或 fsockopen()函数成功打开文件或连接上服务器后所返回的文件标识。

☑　length：可选参数。用于指定读取一行最多的字节数，并返回 length-1 个字节的字符串。如果省略该参数，则 length 的长度默认为 1024 个字节。

（3）fputs()函数是 fwrite()函数的别名，用于将字符串写入到指定的文件中，并可以规定写入字节的大小。该函数的语法如下：

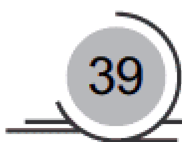
```
int fwrite( resource handle, string string [, int length])
```

fputs()函数的参数说明如表 2.2 所示。

表 2.2　fputs()函数的参数说明

参　数	说　明
handle	必要参数。文件标识指针
string	必要参数。要写入某文件的字符串
length	可省参数。指写入文件的长度，如果省略该参数，则将指定字符串的所有内容写入到文件中

至此，简单的数字计数器的计数部分设计完成，接下来要做的就是在网页中重新输出文本文件中



的数据，即输出网站当前的访问量。代码如下：

```
<?php
//从文本文件中读取统计数据
if(($fp=fopen("counter.txt","r"))==false){          //打开指定的文件
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);                           //读取文件内容
fclose($fp);
echo "数字计数器: " .$counter ;                     //输出访问次数
}
?>
```

2.3　图形数字计数器

所谓图形数字计数器，就是将计数器中的数据以数字图形的形式显示出来，其设计的原理与简单数字计数器是相同的，唯一不同的地方就是在对数据的输出过程中，数字计数器直接将数据进行输出，而图形数字计数器则将数据转化成数字图形的形式进行输出。其运行结果如图 2.4 所示。



图 2.4　图形数字计数器

计数器功能的实现方法已经在简单数字计数器中进行了详细的讲解，这里不再赘述。这里主要讲解如何将文本文件中的数据以图形的形式进行输出。

图形数字计数器存储于 mr\02\02\文件夹下，其实现关键步骤如下：

首先，是读取文本文件中的数据，将文本文件中的数据写入到一个变量中。代码如下：

```
<?php session_start();
if(!isset($_SESSION['temp'])){          //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义变量
if(($fp=fopen("counter.txt","r"))==false){
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);               //读取文件中的数据
fclose($fp);                            //关闭文本文件
$counter++;                             //计数器增加 1
$fp=fopen("counter.txt","w");           //以写的方式打开文本文件
fputs($fp,$counter);                    //将新的统计数据增加 1
fclose($fp);
```



```
}                                    //关闭文件
$_SESSION['temp']=1;                 //为$_SESSION[temp]赋一个值
}
?>
```

然后，将读取的数据以图形的形式进行输出，这里设计的是一个 6 位数字的图形计数器，将 0～9 的数字图形存储在根目录下的 images 文件夹中。步骤如下：

（1）通过 strlen()函数获取数据的长度。

（2）输出 6 位数字的图形计数器中的空值，即处理 6 位数字中补位值 0 的输出。首先，应用 str_repeat()函数获取 6 位数字中补位值 0 的个数，然后应用 for 语句，根据补位值 0 的个数（$str），循环输出代表 0 的图形。

（3）将文本文件中存储的数据以图形的形式输出。应用 for 循环语句，根据 strlen()函数获取变量 $counter 的长度，循环输出指定的图形；应用 switch 语句为数据指定不同的图形。

```
$len=strlen($counter);                          //获取字符串的长度
$str=str_repeat("0",6-$len);                    //获取 6-$len 个数字 0
for($i=0;$i<strlen($str);$i++){                 //获取变量$str 的字符串长度
$result=$str[$i];
$result='<img src=images/0.gif>';
echo $result;                                   //循环输出$result 的结果
}
for($i=0;$i<strlen($counter);$i++){             //获取字符串的长度
$result=$counter[$i];                           //获取数据
switch($result){                                //为不同的数据指定不同的图形
//如果值为 0，则输出 0.gif 图片
case "0"; $ret[$i]="0.gif";   break;
case "1"; $ret[$i]="1.gif";   break;
case "2"; $ret[$i]="4.gif";   break;
case "3"; $ret[$i]="3.gif";   break;
case "4"; $ret[$i]="4.gif";   break;
case "5"; $ret[$i]="5.gif";   break;
case "6"; $ret[$i]="6.gif";   break;
case "7"; $ret[$i]="7.gif";   break;
case "8"; $ret[$i]="4.gif";   break;
case "9"; $ret[$i]="9.gif";   break;
}
echo "<img src=images/".$ret[$i].">";           //输出文本文件中存储的数据
}
}
>
```

说明

① strlen()函数用于获取指定字符串的长度。

② str_repeat()函数用于将指定的字符串重复输出。

2.4　GD2 图形计数器

GD2 图形计数器将计数器中存储的数据以 GD2 函数绘制图形的形式进行输出。其运行结果如图 2.5 所示。



图 2.5　GD2 图形计数器

计数器中的数据仍然存储在文本文件中，至于如何存储和读取其中的数据这里不再赘述，直接讲解如何将读取的数据以 GD2 图形的形式进行输出。GD2 图形计数器存储于 mr\02\03 文件夹下，其关键实现步骤如下：

首先，创建一个画布，并指定画布背景的颜色和字体的颜色。通过 imagecreate()函数创建画布，应用 imagecolorallocate()函数来分别定义背景和字体的颜色，其中在定义字体的颜色时应用 rand()函数，通过其在指定的范围内随机选择颜色。

然后，应用 iconv()函数对指定的中文字符进行转换，并应用 imagettftext()函数输出中文字符串。

最后，应用 imagestring()函数输出计数器中存储的数据，并应用 imagepng()函数以 PNG 格式将图像输出到浏览器或文件中，通过 imagedestroy()函数销毁图像。代码如下：

```
<?php
if(($fp=fopen("counter.txt","r"))==false){
      echo "打开文件失败!";
}else{
      $counter=fgets($fp,1024);
      fclose($fp);
      //通过 GD2 函数创建画布
      $im=imagecreate(240,24);
      $gray=imagecolorallocate($im,255,255,255);   //定义背景
      $color =imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));      //定义字体颜色
      //输出中文字符
      $text=iconv("gb2312","utf-8","网站的访问量：");          //对指定的中文字符串进行转换
      $font = "fonts/FZHCJW.TTF";
      imagettftext($im,14,0,20,18,$color,$font,$text);             //输出中文
      //输出网站的访问次数
      imagestring($im,5,160,5,$counter,$color);
      imagepng($im);
      imagedestroy($im);   //以 PNG 格式将图像输出到浏览器
}
?>
```



（1）imagecreate()函数用于创建一个新的图像。该函数的语法如下：

```
resource imagecreate( int x_size, int y_size )
```

该函数用于返回一个图像标识符，参数 x_size、y_size 为图像的尺寸，单位为像素（pixel）。

（2）imagecolorallocate()函数返回一个标识符，代表了由给定的 RGB 成分组成的颜色。该函数的语法如下：

```
int imagecolorallocate( resource image, int red, int green, int blue)
```

☑ image 参数是 imagecreatetruecolor()函数的返回值。

☑ red、green 和 blue 分别是所需要颜色的红、绿和蓝成分。这些参数是 0～255 的整数或者十六进制的 0x00～0xFF。

imagecolorallocate()函数必须被调用以创建每一种用在 image 所代表的图像中的颜色。第一次对 imagecolorallocate()函数的调用会填充背景色。

（3）rand()函数产生一个随机数，返回随机数的值。该函数的语法如下：

```
int rand( int min, int max)
```

如果没有提供可选参数 min 和 max，则返回 0～RAND_MAX 之间的伪随机数；如果提供可选参数，则返回可选参数范围之内的随机数。

（4）iconv()函数用于将一种已知的字符集文件转换成另一种已知的字符集文件。该函数的语法如下：

```
iconv( string in_charset, string out_charset, string str)
```

（5）imagettftext()函数以 TTF（TrueType Fonts）字体向图像中绘制字符串。该函数的语法如下：

```
array imagettftext( resource image, float size, float angle, int x, int y, int color, string fontfile, string text )
```

imagettftext()函数的参数说明如表 2.3 所示。

表 2.3　imagettftext()函数的参数说明

参　数	说　明
image	图像资源
size	字体大小。根据 GD 版本不同，应该以像素大小（GD1）或点大小（GD2）指定
angle	字体的角度，顺时针计算，0 度为水平，也就是 3 点钟的方向（由左到右），90 度则为由下到上的文字
x	文字的 x 坐标。设定第一个字符的基本点
y	文字的 y 坐标。设定字体基线的位置，不是字符的最底端
color	文字的颜色
fontfile	字体的文件名称，也可以是远端的文件
text	字符串内容

（6）imagestring()函数实现水平的绘制一行字符串。语法如下：

```
bool imagestring( int im, int font, int x, int y, string str, int col )
```

imagestring()函数用 col 颜色将字符串 str 绘制到 im 所代表的图像的（x，y）坐标处（这是字符串左上角坐标，整幅图像的左上角为（0，0））。如果 font 是 1、2、3、4 或 5，则使用内置字体。

至此，GD2 图形计数器讲解完毕。上述介绍的这 3 种计数器都是从输出的方式上进行区别的，其统计的方式是完全相同的，下面将要继续介绍几种完全不同的计数器。

2.5　数据库数字计数器

数据库计数器，顾名思义，就是将统计的数据存储到数据库中，不但可以统计出一个数字值，而且还可以记录更多的信息（例如，IP 地址、访问时间等），这些都可以存储在数据库中。数据库计数器相对于其他几种方法而言，它的功能是最为强大的。其运行结果如图 2.6 所示。

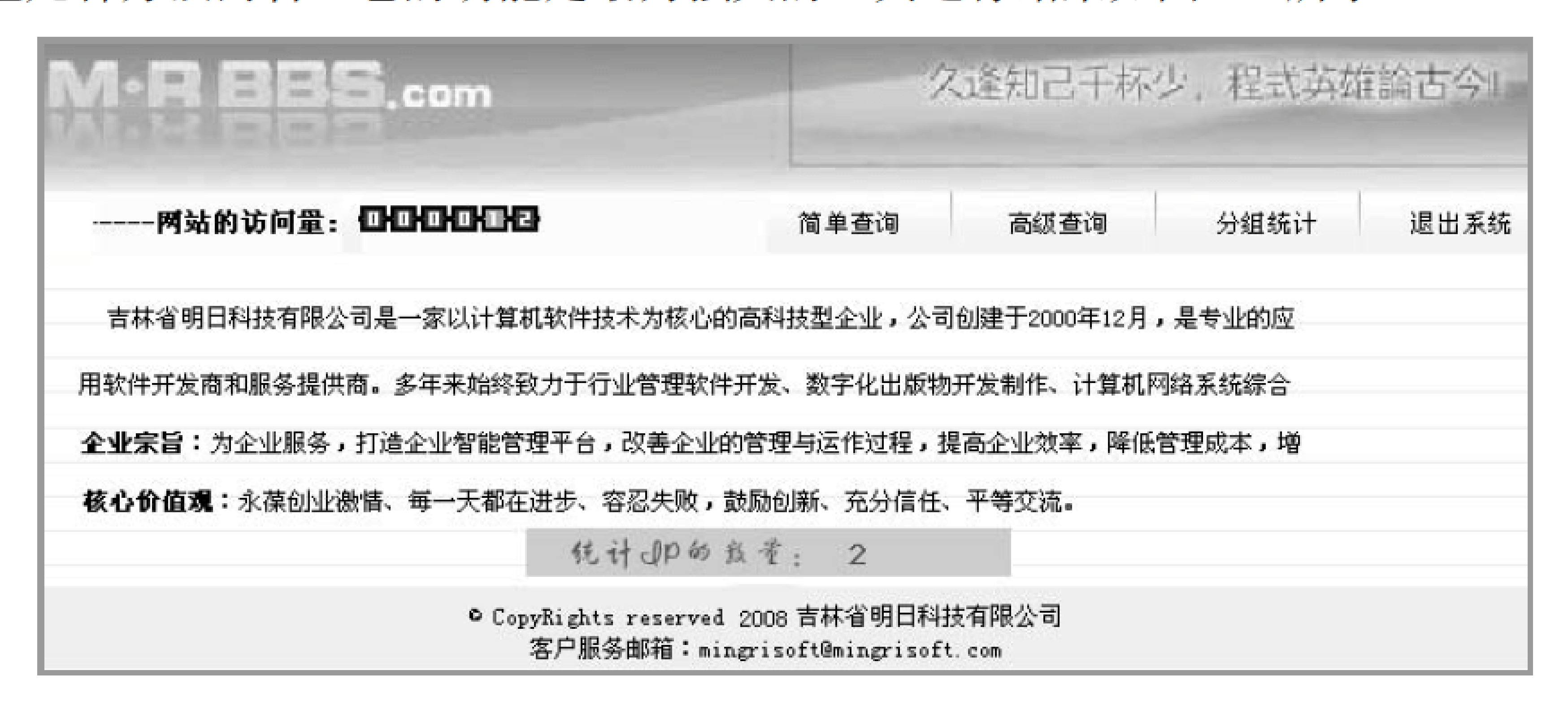


图 2.6　数据库数字计数器

既然数据库计数器需要数据库的支持，所以首先要创建一个数据库，并在该数据库中创建一个数据表用于存储数据。创建的数据表如图 2.7 所示。

服务器: localhost ▸ 数据库: db_counter ▸ 表 : tb_count04

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

	字段	类型	整理	属性	Null	默认	额外	操作
☐	id	int(10)			否		auto_increment	
☐	counts	varchar(50)	gb2312_chinese_ci		否			
☐	data1	varchar(50)	gb2312_chinese_ci		否			
☐	data2	varchar(50)	gb2312_chinese_ci		否			
☐	ip	varchar(50)	gb2312_chinese_ci		否			

图 2.7　创建的数据表

然后连接数据库，连接数据库文件（mr\02\04\conn\conn.php）的代码如下：

```
<?php
$id=mysql_connect('localhost','root','111');//连接数据库服务器
mysql_select_db('db_counter',$id);          //连接 db_counter
mysql_query("set names gb2312");            //指定数据库中字符串的编码格式为 gb2312
?>
```



注意 在连接数据库的过程中，要注意数据库服务器、用户名和密码的使用，还有连接的数据库要正确，以及对数据库中字符串编码格式的转换。

最后实现网站访问信息的存储，并且将访问量和 IP 访问量以图形和图像的形式进行输出。数据库数字计数器存储于 TM\02\04\文件夹下，其关键实现步骤如下：

（1）初始化一个 Session 变量，与数据库建立连接，获取当前时间和客户端的 IP 地址，并判断 Session 变量的值是否为空。代码如下：

```
<?php
session_start();
include("conn/conn.php");
$data1=date("Y-m-d h:m:s");          //获取当前访问时间
$data2=date("Y-m-d");                //获取当前访问时间
$ip=getenv('REMOTE_ADDR');           //获取客户端的 IP 地址
if($_SESSION['temp']==""){           //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义变量
```

（2）通过数据库存储网站被访问的数据，执行 select 查询，判断当前获取的 IP 地址在数据库中是否已经存在。如果数据库中存在该 IP 地址，则执行 update 更新语句，将该 IP 地址对应的数据的访问量增加 1，并且更新访问的时间；如果数据库中不存在该 IP 地址，则执行 insert 添加语句，将该 IP 地址的数据添加到数据库中。代码如下：

```
$query=mysql_query("select * from tb_count04 where ip='$ip'");
$result=mysql_num_rows($query);
if($result>0){
$query_1=mysql_query("update tb_count04 set counts=counts+1,data1='$data1',data2='$data2' where ip='$ip'");
}else{
$query="insert into tb_count04(counts,data1,data2,ip)values('1','$data1','$data2','$ip')";
$result=mysql_query($query);
}
```

（3）将 Session 变量赋值为 1，防止因页面刷新导致重复计数。代码如下：

```
$_SESSION['temp']=1;          //登录以后，$_SESSION[temp]的值不为空，给$_SESSION[temp]赋值 1
}
?>
```

至此，一次网站访问量的统计结束。

2.6　网站访问量统计分析

计数器中的有些功能已经在前面的章节中讲解过了，例如，采用数据库来统计网站的访问量，通过 Session 来防止网站的重复计数，应用图形数字计数器来输出网站当前的访问量。下面介绍如何对网站访问量的数据进行统计分析。这里主要应用 Jpgraph 类创建柱形图来对网站访问量的结果进行分析，

其运行效果如图 2.8 所示。

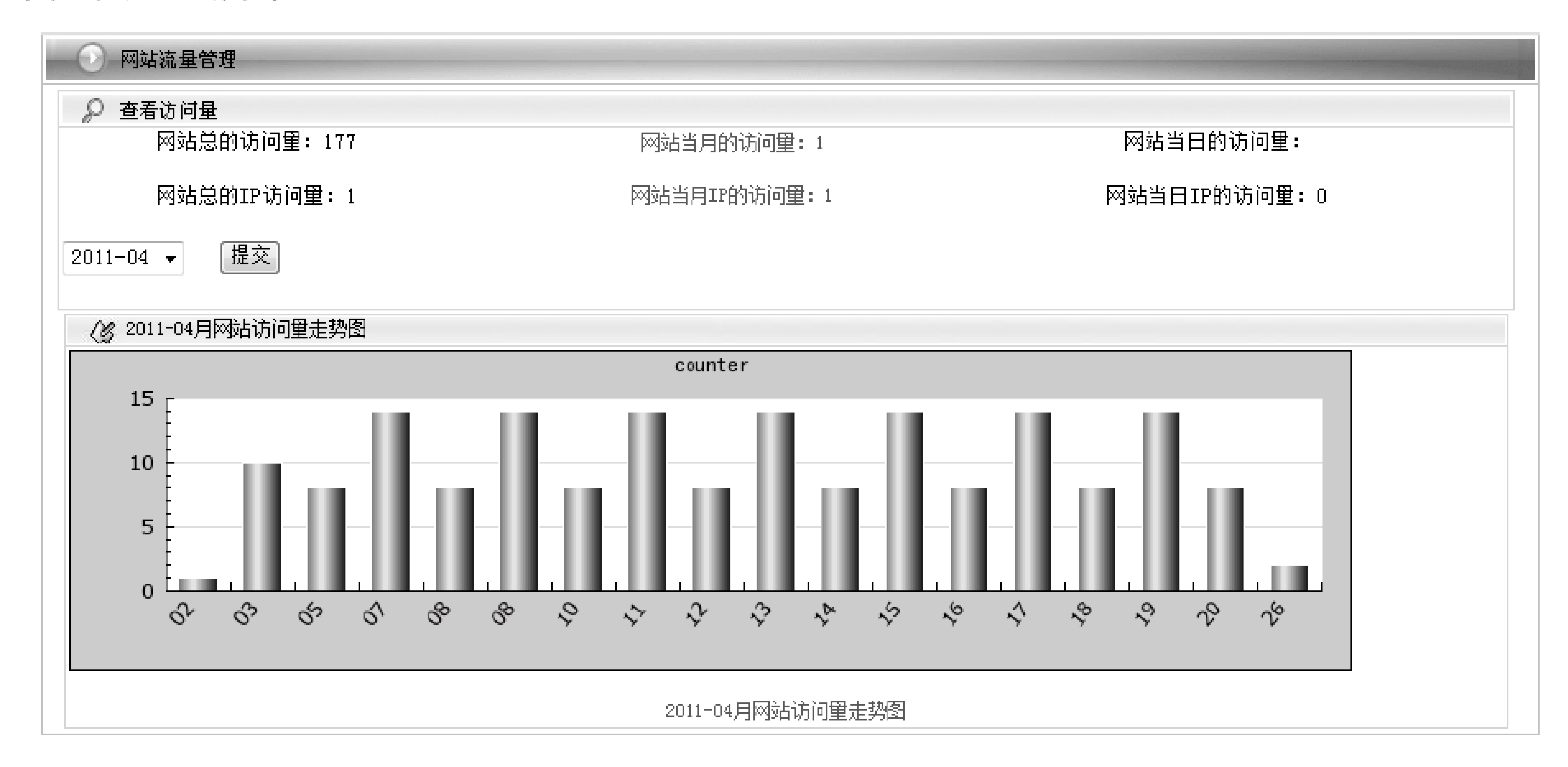


图 2.8　对网站访问量的结果进行分析

2.6.1　通过数据库记录网站访问量

有关数据库统计网站访问量的具体方法已经在 2.5 节中进行了详细讲解，这里不再赘述。输出数据库中存储的网站访问量应用的是 2.3 节的图形数字计数器方法。其关键代码如下：

```
<?php
//以图形的形式输出数据库中的记录数
$querys="select sum(counts) as ll from tb_count10 ";         //查询数据库中总的访问量
$results=mysql_query($querys,$id);
$fwl=mysql_result($results,0,'ll');
echo "----------";
//对补位数字 0 的处理
$len=strlen($fwl);                                            //获取字符串的长度
$str=str_repeat("0",6-$len);                                  //获取 6-$len 个数字 0
for($i=0;$i<strlen($str);$i++){                               //获取变量$str 的字符串长度
$resultes=$str[$i];
$resultes='<img src=images/0.gif>';
echo $resultes;                                               //循环输出$resultes 的结果
}
//对数据库中数据的处理
for($i=0;$i<strlen($fwl);$i++){                               //获取字符串的长度
$result=$fwl[$i];
switch($result){
//如果值为 0，则输出 0.gif 图片
case "0"; $ret[$i]="0.gif";break;
case "1"; $ret[$i]="1.gif";break;
case "2"; $ret[$i]="2.gif";break;
case "3"; $ret[$i]="3.gif";break;
case "4"; $ret[$i]="4.gif";break;
case "5"; $ret[$i]="5.gif";break;
```



```
case "6"; $ret[$i]="6.gif";break;
case "7"; $ret[$i]="7.gif";break;
case "8"; $ret[$i]="8.gif";break;
case "9"; $ret[$i]="9.gif";break;
}
echo "<img src=images/".$ret[$i].">";                    //输出访问次数
}
?>
```

网站访问量的输出结果如图 2.9 所示。

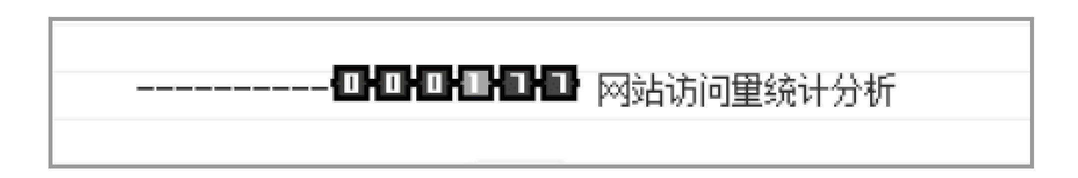


图 2.9　输出网站访问量

2.6.2　网站访问量统计分析

在图 2.9 的网站访问量输出结果中，有一个“网站访问量统计分析”超链接，它链接到 indexs.php 文件中，在此文件中完成对网站访问量的统计操作，其统计的内容如图 2.10 所示。



图 2.10　网站访问量统计

（1）连接数据库，调用指定的数据库连接文件。代码如下：

```
<?php   include("conn/conn.php"); ?>
```

（2）输出网站总的访问量以及总的 IP 访问量。应用 select 语句和 sum()函数，从数据表中读取 counts 字段总的数据，获取网站总的访问量。代码如下：

```
<td width="240" height="25" align="center" valign="middle">
<?php
$query_1="select sum(counts) as countes from tb_count10 ";
//查询数据库中总的访问量
$result_1=mysql_query($query_1);
$countes_1=mysql_result($result_1,0,'countes');
echo "<p class='STYLE1'>网站总的访问量：$countes_1</p>";
```

（3）获取网站总的 IP 访问量。从数据表中读取数据，应用 mysql_fetch_array()函数获取 ip 字段对应的 IP 地址，并将获取到的值赋给一个数组，通过 array_unique()函数去除数组中重复的值，最终获取到网站总的 IP 访问量。代码如下：

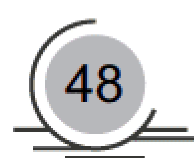
```
//查询数据库中总的 IP 访问量
$query_2="select * from tb_count10 ";
$result_2=mysql_query($query_2);
while($myrow=mysql_fetch_array($result_2)){
$counts_2[]=$myrow['ip'];                          //将获取的 ip 值赋给变量
}
$results_2=array_unique($counts_2);                //去除数组中重复的值
$countes_2=count($results_2);                      //获取数组中值的数量，即总的 IP 访问量
echo "<p class='STYLE1'>网站总的 IP 访问量：$countes_2</p>";
?></td>
```

（4）获取网站当月的访问量和当月的 IP 访问量。使用的方法与获取网站总的访问量和总的 IP 访问量的方法是相同的。区别之处是 select 语句中增加了一个查询条件“data2='".substr(date("Y-m-d"),0,7)."'”，即查询数据库中符合当前月份的数据。代码如下：

```
<td width="240" align="center" valign="middle">
<?php
//查询数据库中当月总的访问量
$query_3="select sum(counts) as countes from tb_count10 where data2='".substr(date("Y-m-d"),0,7)."'";
$result_3=mysql_query($query_3);
$countes_3=mysql_result($result_3,0,'countes');
echo "<p class='STYLE2'>网站当月的访问量：$countes_3</p>";
//查询数据库中当月的 IP 访问量
$query_4="select * from tb_count10 where data2='".substr(date("Y-m-d"),0,7)."'";
$result_4=mysql_query($query_4);
$counts_4=array();
while($myrow_4=mysql_fetch_array($result_4)){
$counts_4[]=$myrow_4['ip'];                        //将获取的 ip 值赋给变量
}
$results_4=array_unique($counts_4);                //去除数组中重复的值
$countes_4=count($results_4);                      //获取数组中值的数量，即总的 IP 访问量
echo "<p class='STYLE2'>网站当月 IP 的访问量：$countes_4</p>";
?></td>
```

（5）获取网站当日的访问量和当日的 IP 访问量。其使用的方法这里不再赘述，指定的查询条件是“data1='".date("Y-m-d")."'”。程序关键代码如下：

```
<td width="248" align="center" valign="middle">
<?php
//查询数据库中当日总的访问量
$query_5="select sum(counts) as countes from tb_count10 where data1='".date("Y-m-d")."' ";
$result_5=mysql_query($query_5);
$countes_5=mysql_result($result_5,0,'countes');
echo "<p class='STYLE3'>网站当日的访问量：$countes_5</p>";
```



```
//查询数据库中当日的 IP 访问量
$query_6="select * from tb_count10 where data1='".date("Y-m-d")."'";
$result_6=mysql_query($query_6);
$counts_6=array();
while($myrow_6=mysql_fetch_array($result_6)){
$counts_6[]=$myrow_6['ip'];                              //将获取的 ip 值赋给变量
}
if(is_array($counts_6)){
$results_6=array_unique($counts_6);                      //去除数组中重复的值
$countes_6=count($results_6);                            //获取数组中值的数量，即总的 IP 访问量
echo "<p class='STYLE3'>网站当日 IP 的访问量：$countes_6</p>";
}
<d>
```

2.6.3　柱形图分析网站月访问量比重

在 indexs.php 文件中，通过下拉列表框选择指定的月份，单击“提交”按钮，将在本页以柱形图的形式输出网站指定月份的访问量比重。

首先创建表单，提交指定要分析的月份。其关键代码如下：

```
<form name="form1" method="post" action="indexs.php">
<select name="select">
<?php
$que="select distinct data2 from tb_count10";          //查询数据库中月份的数据
$res=mysql_query($que,$id);
while($myrow=mysql_fetch_array($res)){                  //循环输出月份数据
?>
<option value="<?php echo $myrow['data2'];?>"><?php echo $myrow['data2'];?></option>
<?php
}
?>
</select>  
<input type="submit" name="Submit" value="提交">
</form>
```

然后在本页中获取表单提交的月份数据，以表单提交的数据为条件进行查询，将从数据库中读取的数据存储到指定的数组中，并将数组重新生成新字符串，将新字符串作为超链接的参数值传递到 stat.php 文件中。关键代码如下：

```
<?php
if(isset($_POST['select'])){                            //判断提交的数据是否为真
$query="select counts,data1 from tb_count10 where data2='".$_POST['select']."' order by data1 ";//定义 SQL 语句
$result=mysql_query($query);                            //执行查询语句
$results=array();                                       //定义数组
$datas=array();                                         //定义数组
while($myrow=mysql_fetch_array($result)){               //循环输出查询结果
$results[]=$myrow['counts'];                            //将查询结果存储到数组中
```

```
$lmbs=implode(",",$results);                                       //将数组重新生成一个字符串
$datas[]=substr($myrow['data1'],8,2);                              //截取时间数据，并存储到数组中
$das=implode(",",$datas);                                          //将数组重新生成一个字符串
}
?>
<table width="902" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#D5D5D5">
<tr>
<td height="18" background="images/mysql_9.jpg" bgcolor="#FFFFFF" class="STYLE4">
<?php echo $_POST['select'];?>月网站访问量走势图
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">
<p><img src="stat.php?lmbs=<?php echo $lmbs; ?>&das=<?php echo $das;?>"/></p>
<p align="center" class="STYLE2"><?php echo $_POST['select'];?>月网站访问量走势图</p>
</td>
</tr>
</table>
<?php
}
?>
```

最后，在 stat.php 文件中应用 Jpgraph 类库创建柱形图，以超链接传递的参数值作为数据，生成网站月访问量统计图。stat.php 文件的代码如下：

```
<?php
include ("src/jpgraph.php");                                       //载入类库文件
include ("src/jpgraph_bar.php");
$lmbs=$_GET['lmbs'];                                               //获取超链接传递的参数值
$das=$_GET['das'];
$datay=explode(",",$lmbs);                                         //按照“,”分隔字符串，$data 为获取的数据变量
$datax=explode(",",$das);                                          //按照“,”分隔字符串，$data 为获取的数据变量
$graph = new Graph(800,200,"auto");                                //创建图像
$graph->img->SetMargin(60,20,30,50);                               //设置图像边框间距
$graph->SetScale("textlin");                                       //定义坐标刻度类型
$graph->SetMarginColor("lightblue");                               //定义图像颜色
$graph->title->Set('counter');                                     //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                        //设置标题字体
$graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL,10);                   //设置 X 轴的字体
$graph->yaxis->SetFont(FF_VERDANA,FS_NORMAL,10);                   //设置 Y 轴的字体
$graph->xaxis->SetTickLabels($datax);                              //设置 X 轴输出的数据
$graph->xaxis->SetLabelAngle(50);                                  //设置输出文字大小
$bplot = new BarPlot($datay);                                      //实例化图像创建类
$bplot->SetWidth(0.6);                                             //设置柱形图的输出大小
$bplot->SetFillGradient("navy","#FFFF00",GRAD_LEFT_REFLECTION);  //设置图像的类型和填充颜色
$bplot->SetColor("white");                                         //设置图像边框颜色
$graph->Add($bplot);                                               //添加数据
$graph->Stroke();                                                  //生成图像
?>
```



2.7　技 术 提 炼

网页计数器的设计方法很多，应用不同方法设计的计数器的实效性也不尽相同。下面对网页计数器开发中应用到的技术进行提炼。

2.7.1　防止重复计数

在开发网页计数器的程序中，一个最为关键的问题就是如何防止网站的重复计数。如果一个计数器不能控制重复计数，那么这个计数器将没有任何意义。根据不同的计数器的设计方法，总结出两种控制重复计数的方法。

1．通过 Session 防止重复计数

该方法可以应用到任意一个计数器中实现控制重复计数的功能。其实现的原理是：在当前页被访问时，初始化一个 Session 变量，并赋给其一个空值；然后判断 Session 变量的值是否为空，如果为空，则将计数器的值增加 1，并且为 Session 变量赋值为 1。此时，在当前页中，Session 变量的值已经不为空，无论如何刷新，Session 变量的值都不会改变，所以计数器的值也不会增加。关键代码如下：

```
<?php
//使用文本存储数据
if(!isset($_SESSION['temp'])){   //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义变量
if(($fp=fopen("counter.txt","r"))==false){
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);                    //读取文件中数据
fclose($fp);                                 //关闭文本文件
$counter++;                                  //计数器增加 1
$fp=fopen("counter.txt","w");                //以写的方式打开文本文件
fputs($fp,$counter);                         //将新的统计数据增加 1
fclose($fp);
}                                            //关闭文本文件
$_SESSION['temp']=1;                         //计数器的值增加后，为$_SESSION[temp]赋值 1
}
//从文本文件中读取统计数据
if(($fp=fopen("counter.txt","r"))==false){
echo "打开文件失败!";
}else{
$counter=fgets($fp,1024);
fclose($fp);
}
?>
```

注意 Session 变量防止重复计数的缺点是只对当前页面有效，当重新打开该页面时，计数器的值仍然会增加。

2．通过 IP 防止重复计数

通过 IP 防止重复计数是最准确的方法，它完全是通过客户端机器的 IP 来防止重复计数，不会有任何时间和页面的限制，只要是已经访问过的 IP，计数器的值就不会增加。

通过 IP 防止重复计数必须要有数据库的支持，服务器会将每个访问过本网站的客户端的 IP 存储到数据库中。

当客户端的机器访问本网站时，服务器将会首先获取该客户端的 IP 地址，并且与数据库中存储的 IP 地址进行比较，如果数据库中已经存在此 IP，则计数器的值将不会增加；否则将计数器的值加 1，并且将该 IP 地址存储到数据库中。关键代码如下：

```
<?php
session_start();
include("conn/conn.php");
$data1=date("Y-m-d h:m:s");                //获取当前访问时间
$data2=date("Y-m-d");                      //获取当前访问时间
$ip=getenv('REMOTE_ADDR');                 //获取客户端 IP 地址
//使用数据库存储数据
$query=mysql_query("select * from tb_count04 where ip='$ip'");
$result=mysql_num_rows($query);
if($result>0){
echo "该 IP 地址已经存在";
}else{
$query="insert into tb_count04(counts,data1,data2,ip)values('1','$data1','$data2','$ip')";
$result=mysql_query($query);
}
?>
```

通过 IP 不但可以防止重复计数，而且可以统计出网站被同一 IP 访问的次数。

2.7.2　通过 array_unique()函数获取网站当月 IP 访问量

通过 array_unique()函数去除数组中重复的值，从而实现统计网站当月 IP 的访问量。其中主要应用了 array_unique()函数和 count()函数。

（1）array_unique()函数

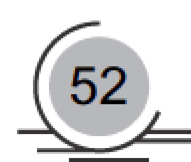
该函数将值作为字符串排序，然后对每个值只保留第一个键名，忽略所有后面的键名。语法如下：

```
array array_unique( array array)
```

（2）count()函数

该函数返回数组中的单元数目，用来计算数组中值的个数。语法如下：

```
int count( mixed array [, int mode])
```



参数 array 为指定的数组；参数 mode 为可选参数，如果 mode 的值设置为 COUNT_RECURSIVE（或 1），该函数将递归地对数组计数。对计算多维数组的所有单元尤其有用。此参数的默认值是 0。

在获取网站当月 IP 的访问量之前，先来看一下存储网站访问量数据的数据表的结构，如图 2.11 所示。

服务器: localhost ▸ 数据库: db_counter ▸ 表 : tb_count10

浏览　结构　SQL　搜索　插入　导出　Import　操作　清空　删除

	字段	类型	整理	属性	Null	默认	额外				操作
☐	tb_id	int(10)			否		auto_increment				
☐	counts	varchar(50)	gb2312_chinese_ci		否						
☐	data1	varchar(50)	gb2312_chinese_ci		否						
☐	data2	varchar(50)	gb2312_chinese_ci		否						
☐	ip	varchar(50)	gb2312_chinese_ci		否						

图 2.11　网站访问量统计数据表

在该数据表中，包括 5 个字段，分别是 id；网站访问量 counts；访问时间 data1，完整的时间年月日；访问时间 data2，只包括年和月；网站的 IP 地址 ip。

要获取网站当月 IP 访问量，其查询的条件必须从时间 data2 着手，查询出当前时间符合 data2 的数据。代码如下：

```
<?php
$query_4="select * from tb_count10 where data2='".substr(date("Y-m-d"),0,7)."'";
     $result_4=mysql_query($query_4);
```

技巧　应用 substr()函数可以获取当前日期中年和月的值，将获取的值与数据表中 data2 字段的值进行比较，从中读取出符合条件的数据。

然后，将获取到的 IP 地址写入到一个数组中。代码如下：

```
while($myrow_4=mysql_fetch_array($result_4)){
     $counts_2[]=$myrow['ip'];                    //将获取的 ip 的值赋给变量
}
```

最后，先应用 array_unique()函数，去除数组中重复的值，再统计出数组中元素的数量，即总的 IP 访问量。

```
     $results_4=array_unique($counts_4);          //去除数组中重复的值
     $countes_4=count($results_4);                //获取数组中值的数量，即总的 IP 访问量
?>
```

2.7.3　将数组中的数据写入到字符串中

将数组中的数据写入到字符串中，主要应用的是 current()函数和 implode()函数。

（1）current()函数

该函数返回数组中的当前单元。每个数组中都有一个内部的指针，初始指向插入到数组中的第一

个单元。该函数返回当前被内部指针指向的数组单元的值，并不移动指针。如果内部指针指向超出了数组的最大单元，将返回 false。语法如下：

```
mixed current( array array)
```

（2）implode()函数

该函数将数组的内容组合成一个字符串。语法如下：

```
string implode(string glue, array pieces)
```

参数 glue 是字符之间的分隔符号。

这里实际是将指定月份每天的数据写入到字符串中，实现对网站指定月份的访问量进行分析。要完成该操作，首先要以指定的月份为查询条件，查询出指定月份的网站访问量的数据。代码如下：

```
<?php
	$query="select counts,data1 from tb_count10 where data2='".$_POST['select']."' order by data1 ";
	$result=mysql_query($query);
```

然后通过 while 语句，循环读取其中的数据，并且应用 current()函数将数组中当前的元素写入到数组$results 中。

最后应用 implode()函数，将数组$results 中的数据写入到一个字符串变量$lmbs 中，以“,”为分隔符，完成将指定月份每天的数据写入到字符串中的操作。代码如下：

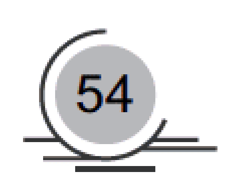
```
	while($myrow=mysql_fetch_array($result)){
		$results[]=$myrow['counts'];
		$lmbs=implode(",",$results);
	}
?>
```

2.7.4 动态生成柱形图分析网站访问量

动态生成柱形图分析网站访问量，主要应用 Jpgraph 类库根据从数据库中读取的数据生成柱形图，完成对网站访问量的分析。其前提是必须在项目中载入 Jpgraph 类库，由于其属于第三方的组件，在本书的光盘源码中没有提供，所以需要读者自行下载，并且将下载的 src 文件夹复制到项目 mr\02\05 文件夹下，然后才可以完成动态生成柱形图分析网站访问量的操作。

2.8 本 章 小 结

本章通过网页计数器带领读者详细讲解一个计数器的使用流程。同时，在网页计数器的使用过程中，采用了 Session 技术、Cookie 技术，使整个页面设计思路更加清晰。通过本章的学习，读者不仅可以了解一般网站的访问量的统计，而且可以更加熟悉 Session、Cookie 的使用。



第 3 章

上传与下载模块

（PHP+Ajax+MySQL 实现）

文件的上传与下载是一个完整的网站应该具有的功能之一，如果网站中不具备这样的功能，给浏览者的感觉是这个网站的开发不是很完整。本章将全面介绍 PHP 的上传与下载技术，从 php.ini 文件的配置，到上传函数的使用，到最后的动手实践。

通过阅读本章，可以学习到：

- **文件目录及文件的相关操作**
- **文件的上传**
- **文件的下载**
- **文件重名自动检测**
- **利用下载码实现文件搜索**
- **如何限制用户的空间使用大小**
- **文件上传的相关配置**

3.1 上传与下载模块概述

3.1.1 模块概述

使用过 ASP 的读者一定会对 ASP 上传记忆犹新。要在 ASP 中实现上传，或者使用 Active 组件上传，或者编写一大段代码实现无组件上传。这对于新手来说有点望而生畏，甚至对于成手也有一定的难度。PHP 在这方面要比 ASP 方便得多，只要一个上传函数就能够完全实现。如果是大文件上传或下载，也只是在 php.ini 配置文件中设置几个参数值即可。在本模块中将对 PHP 的上传和下载功能的实现进行系统全面的讲解。

3.1.2 功能结构

为了读者更好地学习，下面给出上传下载模块的功能结构流程图。

用户在上传下载模块中通过注册为本模块用户，然后登录模块进行相关操作。用户正确登录后，系统将自动判别访客的级别，然后根据不同的级别分配不同的权限。用户未登录时，身份为游客，可通过下载码下载未公开文件。用户成功登录后，根据级别的不同，可以执行上传新文件、删除已上传个人文件和更改文件公布状态等操作。

上传与下载模块的功能结构如图 3.1 所示。

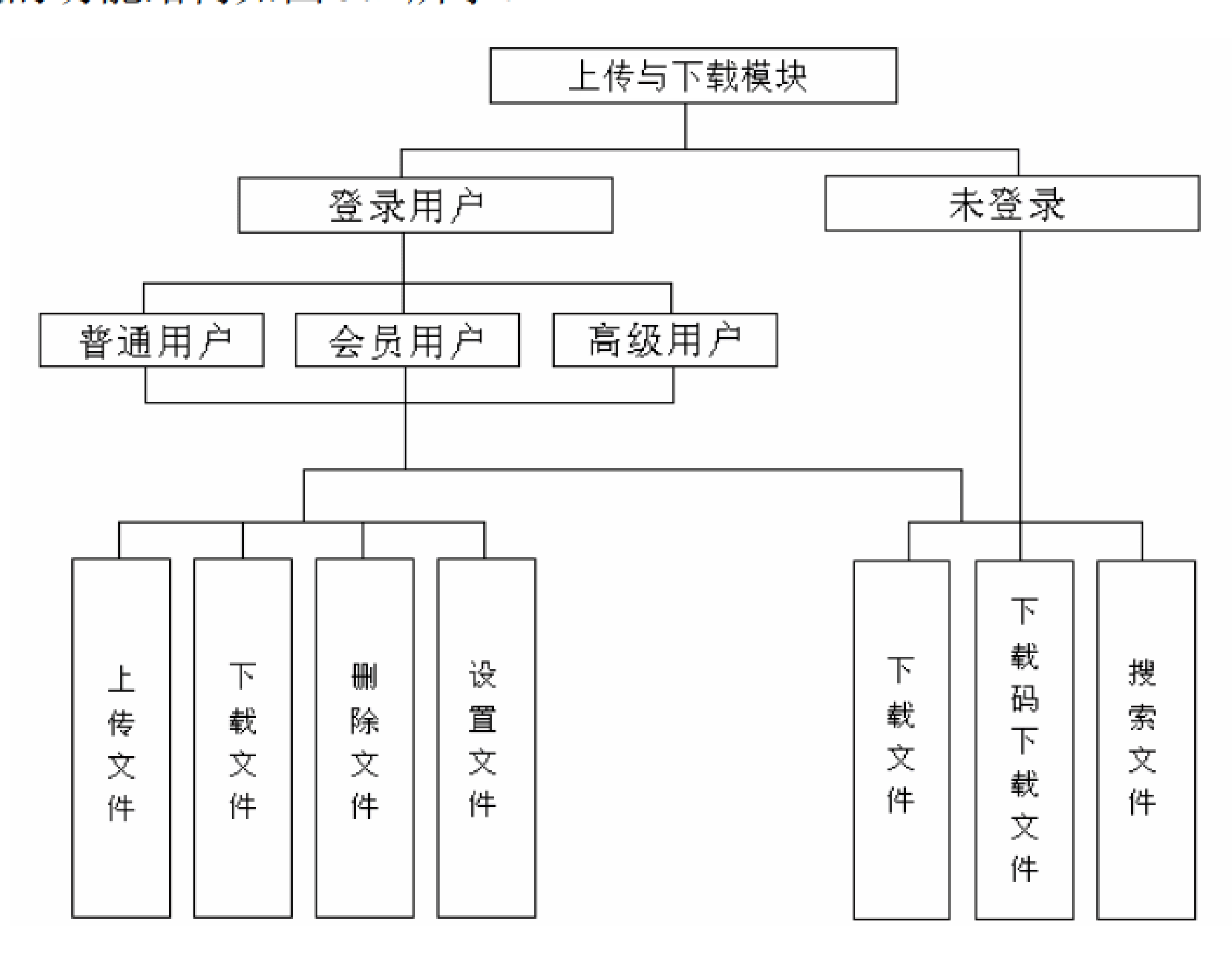
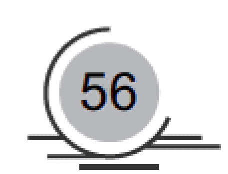


图 3.1 上传下载模块功能结构图

3.1.3 程序预览

上传与下载模块由多个页面组成，下面列出几个典型的页面，其他的请参看光盘中的程序。





上传与下载模块的登录页面如图 3.2 所示，用户想下载自己上传的文件，则首先执行登录操作，登录成功后才可以在个人信息显示处单击“下载文件”超链接进入下载页。



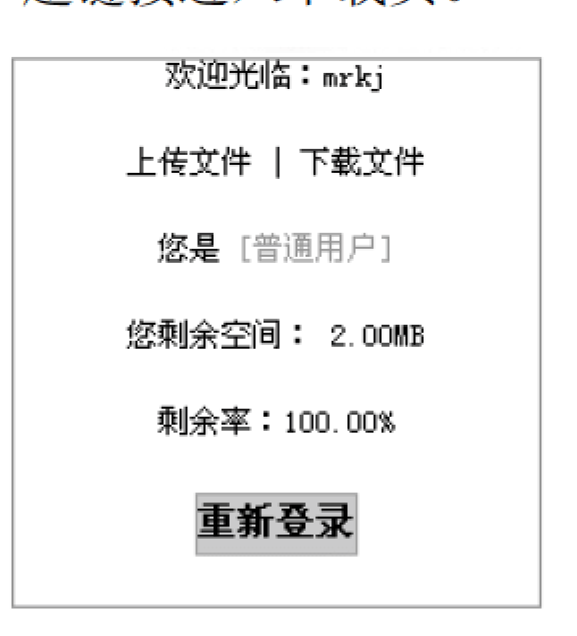


图 3.2　上传下载模块的登录页面

上传与下载模块的主页如图 3.3 所示。



图 3.3　上传下载模块的主页

上传文件页面如图 3.4 所示，该页面主要是上传新文件到系统分配的空间内。

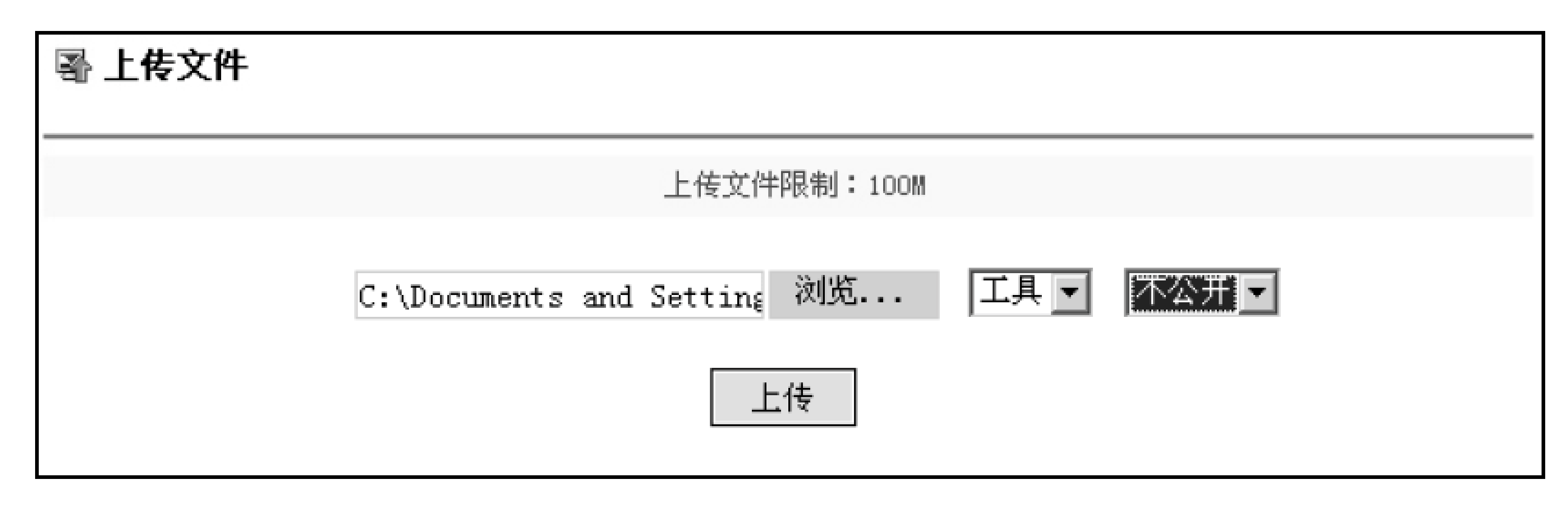
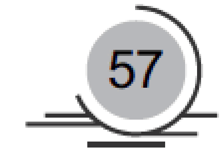


图 3.4　上传文件页面



下载页面如图 3.5 所示，顾名思义，该页面主要是展示已上传文件的相关信息，用户可以通过单击该页面上文件名称的超链接下载该文件。

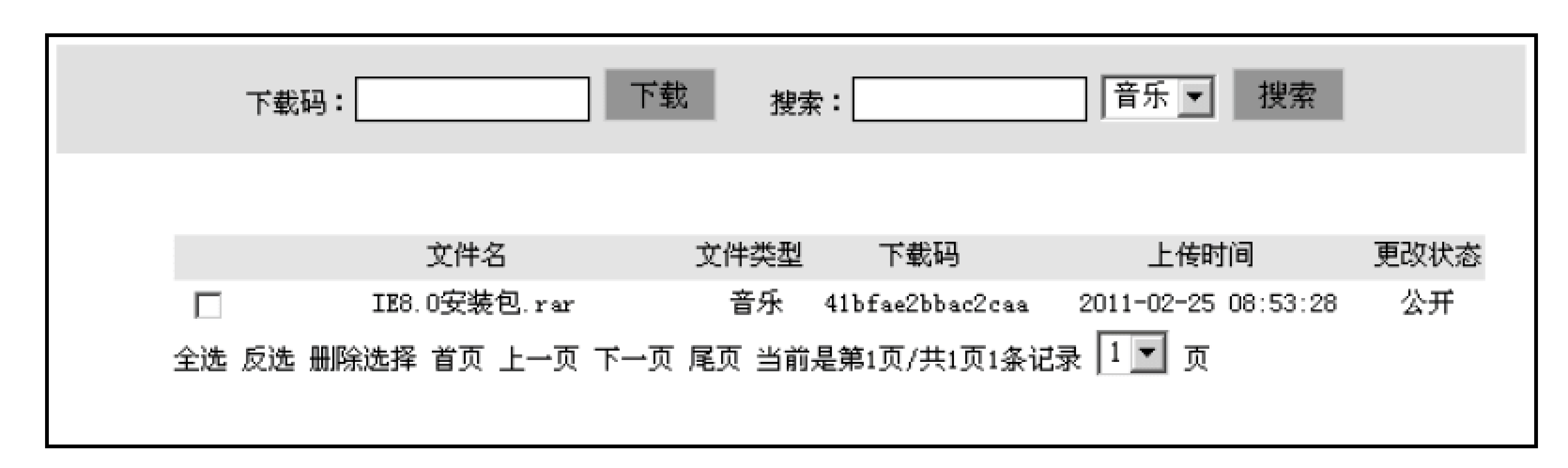


图 3.5　下载页面

当前用户为游客身份时，也可以通过文件的下载码下载对应的上传文件，如图 3.6 所示。

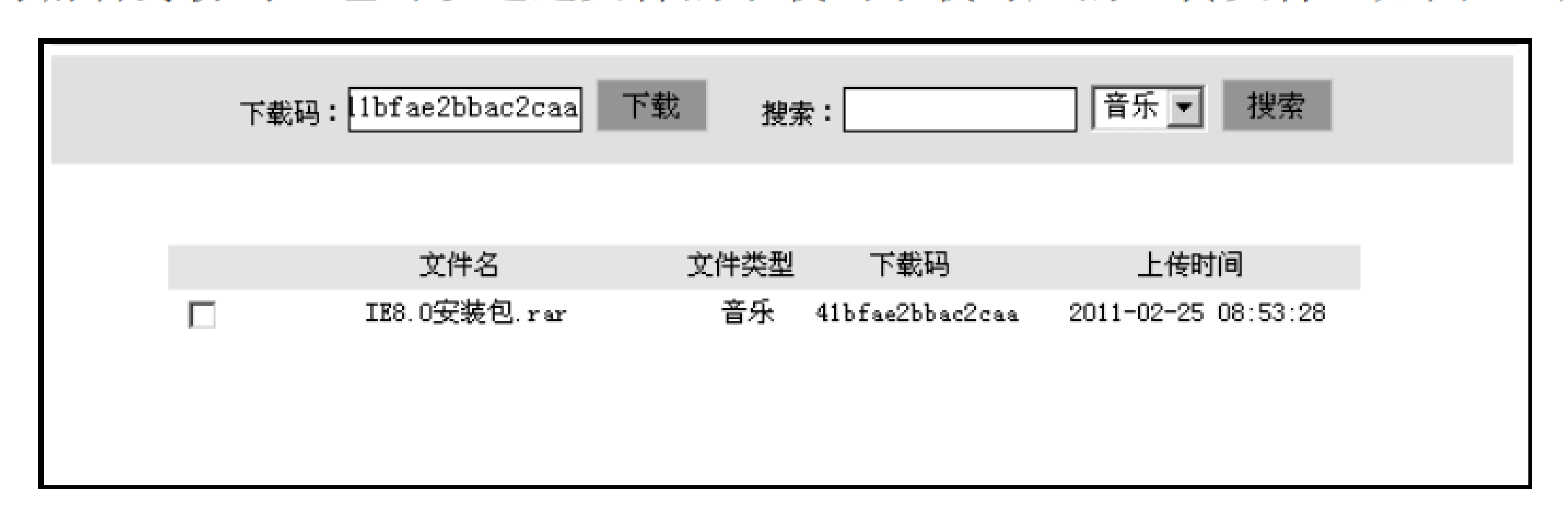


图 3.6　通过下载码下载文件

3.2　数据库设计

本节通过实现一个上传与下载模块，学习上传与下载技术在实际中的应用。首先来看数据库的设计和连接。

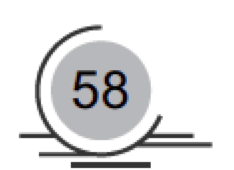
3.2.1　数据库分析

首先定义上传与下载模块使用 MySQL 数据库，名称是 db_netu，其中包含 3 个表，分别是用户列表、上传文件列表和文件类型列表。其中用户列表存储用户的相关信息；上传文件列表是整个上传下载模块的核心数据表，记录了上传下载文件的相关信息；文件类型列表主要记录上传文件的种类。数据库 db_netu 包含的数据表如图 3.7 所示。



表	类型	整理	说明
tb_member	MyISAM	gb2312_chinese_ci	用户列表
tb_upfile	MyISAM	gb2312_chinese_ci	上传文件列表
tb_uptype	MyISAM	gb2312_chinese_ci	文件类型列表

图 3.7　数据库 db_netu 包含的数据表





3.2.2　创建数据表

1．tb_member 表

tb_member 表记录注册用户信息，包括文件名称、密码保护、密保问题、E-mail、用户权限以及空间使用情况等信息。tb_member 表结构及说明如图 3.8 所示。

2．tb_upfile 表

tb_upfile 表记录用户上传的文件信息，包括文件名称、文件类型、下载码等信息。tb_upfile 表结构及说明如图 3.9 所示。



服务器: localhost ▸ 数据库: db_netu ▸ 表 : tb_member

浏览 结构 SQL 搜索 插入 导出 Import 操作

	字段	类型	整理	属性	Null	注释
☐	id	int(4)			否	自动编码
☐	name	varchar(50)	gb2312_chinese_ci		否	文件名称
☐	password	varchar(50)	gb2312_chinese_ci		否	用户密码
☐	question	varchar(50)	gb2312_chinese_ci		否	密码保护
☐	answer	varchar(50)	gb2312_chinese_ci		否	密保问题
☐	email	varchar(200)	gb2312_chinese_ci		否	Email
☐	right	int(1)			否	用户权限
☐	content	bigint(50)			否	空间使用情况

图 3.8　tb_member 表结构及说明



服务器: localhost ▸ 数据库: db_netu ▸ 表 : tb_upfile

浏览 结构 SQL 搜索 插入 导出 Import 操

字段	类型	整理	说明
id	int(4)		自动编号
filename	varchar(200)	gb2312_chinese_ci	文件名称
filepath	varchar(200)	gb2312_chinese_ci	保存路径
filetype	varchar(100)	gb2312_chinese_ci	文件类型
upauthor	varchar(200)	gb2312_chinese_ci	上传用户
uptime	timestamp		上传时间
chkdownnum	varchar(20)	gb2312_chinese_ci	下载码
ispub	int(1)		是否公开
filesize	bigint(20)		文件大小

图 3.9　tb_upfile 表结构及说明

3．tb_uptype 表

tb_uptype 表规定上传文件的类型，用户可以将文件上传到不同的类型中，以方便日后查找。tb_uptype 表结构及说明如图 3.10 所示。



服务器: localhost ▸ 数据库: db_netu ▸ 表 : tb_uptype

浏览 结构 SQL 搜索 插入 导出 Import 操作

字段	类型	整理	额外	说明
id	int(4)		auto_increment	自动编号
genrename	varchar(200)	gb2312_chinese_ci		类型名称

图 3.10　tb_uptype 表结构及说明

3.2.3　数据库连接文件

PHP 连接数据库并非难事，但是在每次应用数据库连接时都重复输入类似代码就会耽误大量的时间，同时也降低了代码的书写效率和可读性。所以，本模块中将数据库的连接方法封装到一个类中，这样，在应用数据库连接操作时，调用该类文件即可。conn\conn.php 文件的关键代码如下：

```
<?php
class opmysql{
```

```
private $host = 'localhost';                        //服务器地址
private $name = 'root';                             //登录账号
private $pwd = '111';                               //登录密码
private $dBase = 'db_netu';                         //数据库名称
private $conn = '';                                 //数据库连接资源
private $result = '';                               //结果集
private $msg = '';                                  //返回结果
private $fields;                                    //返回字段
private $fieldsNum = 0;                             //返回字段数
private $rowsNum = 0;                               //返回结果数
private $rowsRst = '';                              //返回单条记录的字段数组
private $filesArray = array();                      //返回字段数组
private $rowsArray = array();                       //返回结果数组
//初始化类
function __construct($host='',$name='',$pwd='',$dBase=''){
if($host != '')
$this->host = $host;
if($name != '')
$this->name = $name;
if($pwd != '')
$this->pwd = $pwd;
if($dBase != '')
$this->dBase = $dBase;
$this->init_conn();
}
//链接数据库
function init_conn(){
$this->conn=@mysql_connect($this->host,$this->name,$this->pwd);
@mysql_select_db($this->dBase,$this->conn);
mysql_query("set names gb2312");
}      private $dBase = 'db_netu';                  //数据库名称
//省略部分代码
```

另外，该自定义类中还包含了一些常用的数据库操作方法，具体方法和相关说明请查看光盘中的源程序，这里不再赘述。

3.3 用 户 注 册

3.3.1 注册模块概述

上传与下载模块是一个面向公开用户的个人网络存储空间程序，自助式的用户注册是其功能模块之一，新用户可以通过单击登录模块中“注册”超链接来进入自助式用户申请页面。其运行结果如图 3.11 所示。

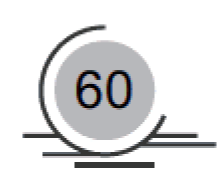




图 3.11　用户注册运行结果

3.3.2　注册页面设计

注册页面的设计没有太多实质性的理论可言，其主要是通过表单来获取用户注册的相关信息，将数据提交到 register_chk.php 文件中，完成用户注册信息的处理。注册页面文件 register.php 的相关代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
?>
<table id="regfm" width="500" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td height="25" colspan="3" align="center" valign="middle">电子相册注册页面</td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">注册名称：</td>
<td width="200" height="25" align="left" valign="middle"> 
<input id="regname" name="regname" type="text" class="txt" /></td>
<td height="25"><div id="namediv" class="regdiv">名称由英文字母和数字及下划线组成</div></td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">注册密码：</td>
<td width="200" height="25" align="left" valign="middle"> 
<input id="regpwd1" name="regpwd1" type="password" class="txt" /></td>
<td height="25"><div id="pwddiv1" class="regdiv">请输入密码</div></td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">确认密码：</td>
<td width="200" height="25" align="left" valign="middle"> 
<input id="regpwd2" name="regpwd2" type="password" class="txt" /></td>
<td height="25"><div id="pwddiv2" class="regdiv">确认密码</div></td>
</tr>
</table>
</table>
<div id="morediv" style="display:non;">
<!--省略部分代码-->
```

```
</table>
</div><center>
<div style=" text-align:center;text-align:center; width:50%; height:25px; line-height:25px;">
<button id="regbtn" class="btn" disabled="disabled">注册</button> 
<button id="morebtn" class="btn">详细信息</button></div></center>
```

3.3.3　Ajax 无刷新技术验证注册信息

新用户在注册新账号时，如果在每一次提交注册信息之后才返回不符合要求的错误信息，不但降低了执行效率，而且也会消磨用户的耐心，这样一来，用户之前填写的所有注册信息会随着操作失败而消失。为了让用户及时得到反馈信息，注册模块采用目前比较流行的 Ajax 技术，对注册信息执行无刷新验证操作。这里以验证注册用户名为例，该过程是通过 js 文件目录中的 register.js 文件来实现的。关键代码如下：

```
function $(id){
return document.getElementById(id);                              //获取标签 ID
}
$('regname').onblur = function(){
name = $('regname').value;
if(cname1 == 'yes'){
xmlhttp.open('get','chkname.php?name='+name,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readyState == 4){
if(xmlhttp.status == 200){
var msg = xmlhttp.responseText;
if(msg == '1'){                                                  //判断用户名是否被占用
$('namediv').innerHTML="<font color=green>恭喜您，该用户名可以使用!</font>";
cname2 = 'yes';
}else if(msg == '2'){
$('namediv').innerHTML="<font color=red>用户名被占用！</font>";
cname2 = '';
}else{
$('namediv').innerHTML="<font color=red>"+msg+"</font>";
cname2 = '';
}
}
}
}
xmlhttp.send(null);
chkreg();
}
}
```

在 register.js 文件中，无刷新调用 chkname.php 文件，对注册用户名进行验证，根据 chkname.php 文件的返回值判断该用户名是否可用。chkname.php 文件的关键代码如下：

```
<?php
```

```
session_start();                                          //开启会话
include_once "conn/conn.php";                             //加载数据库配置文件
$reback = '0';                                            //定义验证返回值
$sql = "select * from tb_member where name='".$_GET['name']."'";
$num = $conne->getRowsNum($sql);                          //执行数据库查询操作
if($num == 1){                                            //判断是否存在匹配结果
$reback = '2';                                            //如果存在，则将验证返回值赋值 2
}else if($num == 0){
$reback = '1';                                            //如果不存在，则将验证返回值赋值 1
}else{
$reback = $conne->msg_error();                            //输出错误信息
}
echo $reback;                                             //输出验证返回值
?>
```

3.3.4　注册信息处理

创建 register_chk.php，获取注册页面 register.php 中提交的注册信息，最终将用户的注册信息添加到指定的数据表中，同时为新注册用户创建一个属于自己的存储文件目录。注册信息处理文件 register_chk.php 的相关代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once 'conn/conn.php';
$reback = '0';
$sql = "insert into tb_member(name,password,question,answer,email)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."',
'".$_GET['answer']."','".$_GET['email']."')";
$num = $conne->uidRst($sql);
if($num == 1){
$_SESSION['name'] = $_GET['name'];
$username=$_SESSION['name'];
$dirname="../upfile/$username";     //创建一个和用户名相同的文件夹
mkdir($dirname,0700);
$reback = '1';
$file = 'index.txt';
$newfile="../upfile/$username/index.php";
if (!copy($file, $newfile)) {
echo "failed to copy $file...\n";
}
}else{
$reback = $conne->msg_error();
}
echo $reback;
?>
```

3.4 用 户 登 录

3.4.1 登录模块概述

上传与下载模块通过用户登录实现个人空间划分、控制上传文件的公开或者隐藏，以及对个人上传文件的管理。用户登录的运行结果如图 3.12 所示。

用户登录模块中包含了用户登录、新用户注册、找回密码等功能，同时输出了用户的权限、剩余空间大小、剩余空间百分比以及上传文件和下载文件的超链接。由于篇幅所限，下面只对用户登录模块中的一些特殊功能进行详细讲解，其完整内容请参考本书光盘中的源代码，这里不再赘述。

图 3.12 登录模块运行结果

3.4.2 用户权限输出以及空间使用计算

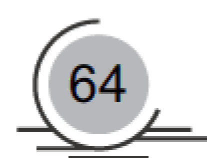
在用户登录后，将输出用户的权限以及用户个人空间的使用情况。下面就介绍如何判断登录用户的权限，以及空间使用情况的计算。

（1）在 index.php 中，根据 Session 变量传递的登录用户名称，从数据库中读取该用户的个人信息，对 tb_member 数据表中 right 字段的值进行判断，进而输出用户的权限。关键代码如下：

```
<?php
$rightsql = "select * from tb_member where name = '".$_SESSION['name']."'";          //查询语句
$rightarr = $conne->getRowsRst($rightsql);                                 //返回查询结果集，判断用户权限
$conne->close_rst();
if($rightarr['right']==2) {                                                 //如果返回查询值为 2，判断为高级用户
echo "<span class=\"vip1\">[高级用户]</span>";
}else if($rightarr['right']==1) {
echo "<span class=\"vip2\">[会员用户]</span>";                             //如果返回查询值为 1，判断为会员用户
}else{
echo "<span class=\"vip3\">[普通用户]</span>";                             //否则判断为普通会员
}
?>
```

（2）计算用户的空间使用情况。首先，从数据库中读取权限字段中的数值，判断当前用户的登录权限，通过用户的权限来设定其使用空间大小；然后，从数据库中读取该用户的空间使用情况，将其分配空间的大小与该用户已使用空间大小做差值；最后，将取得结果进行格式化输出，显示当前登录用户的剩余空间和剩余空间的百分比。相关代码如下：

```
<p>您剩余空间：
<?php
```





```
if($rightarr['right']==2) {                              //如果权限为高级用户，则其分配空间为 5MB
if(5000000-$rightarr['content']<=1000) {                 //判断用户空间是否不足
echo "已满！";
}else {
$formatted=(5000000-$rightarr['content'])/1000000;       //对剩余空间大小进行换算
$forstr=sprintf("%-01.2f",$formatted);                   //对计算的空间大小数值进行格式化
echo $forstr."MB";
echo "<p>剩余率:".sprintf("%-01.2f",((1-$rightarr['content']/5000000)*100))."%</p>";
}                                                        //对用户的剩余空间以百分比的格式化输出
}else if($rightarr['right']==1) {
if(3000000-$rightarr['content']<=1000) {
echo "已满！";
}else{
$formatted=(3000000-$rightarr['content'])/1000000;
$forstr=sprintf("%-01.2f",$formatted);
echo $forstr."MB";
echo "<p>剩余率:".sprintf("%-01.2f",((1-$rightarr['content']/3000000)*100))."%</p>";
}
}else {
if(2000000-$rightarr['content']<=1000) {
echo "已满！";
}else{
$formatted=(2000000-$rightarr['content'])/1000000;
$forstr=sprintf("%-01.2f",$formatted);
echo $forstr."MB";
echo "<p>剩余率：".sprintf("%-01.2f",((1-$rightarr['content']/2000000)*100))."%</p>";
}
}
?>
```

3.4.3　为用户划分个人空间文件目录

为注册的每一个用户创建一个和其注册名同名的个人文件目录，这样不但可以方便管理员对个人文件的管理，同时也避免了所有用户上传文件到同一文件目录下的弊端。

创建个人文件目录的操作在用户的注册处理页 register_chk.php 中完成。首先获取表单提交的用户注册信息，然后执行添加语句将注册信息添加到指定的数据表，最后通过 mkdir()函数在服务器指定的目录下创建一个与用户名相同的文件夹。关键代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once 'conn/conn.php';
$reback = '0';
$sql = "insert into tb_member(name,password,question,answer,email)
values('".trim($_GET['name'])."','".md5(trim($_GET['pwd']))."','".$_GET['question']."',
'".$_GET['answer']."','".$_GET['email']."')";
$num = $conne->uidRst($sql);
if($num == 1){
```

```
$_SESSION['name'] = $_GET['name'];
$username=$_SESSION['name'];
$dirname="../upfile/$username";                    //设置文件夹的路径及名称
mkdir($dirname,0700);                              //创建一个和用户名相同的文件夹
$reback = '1';
```

3.4.4 防止用户文件目录的非法访问

某些用户可能通过不正当方式获取其他用户文件夹的地址，这样就可以直接通过地址访问其他用户文件夹，获得相关的内容并下载其他用户文件。为了避免这一弊端，在为每个用户创建自己文件夹的同时，在自己的文件夹下创建了一个 index.php 文件，以此来屏蔽直接通过地址栏访问文件夹下内容的方式。

首先，在 login 文件夹下创建一个名为 index.txt 的文本文档。代码如下：

```
<?php
  echo "请使用正确的访问路径！";
?>
<p>
单击<a href="../../index.php">这里</a>返回                //单击返回首页
</p>
```

然后，在用户注册的处理页 register_chk.php 中，在完成用户注册信息的添加之后，将 index.txt 文本文件复制到新创建的用户个人文件夹下，并且命名为 index.php。关键代码如下：

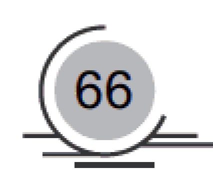
```
$file = 'index.txt';
$newfile="../upfile/$username/index.php";          //设置复制新路径名称
if (!copy($file, $newfile)) {                      //将 index.txt 复制到新路径
    echo "failed to copy $file...\n";
}
}else{
$reback = $conne->msg_error();
}
echo $reback;
?>
```

3.5 找 回 密 码

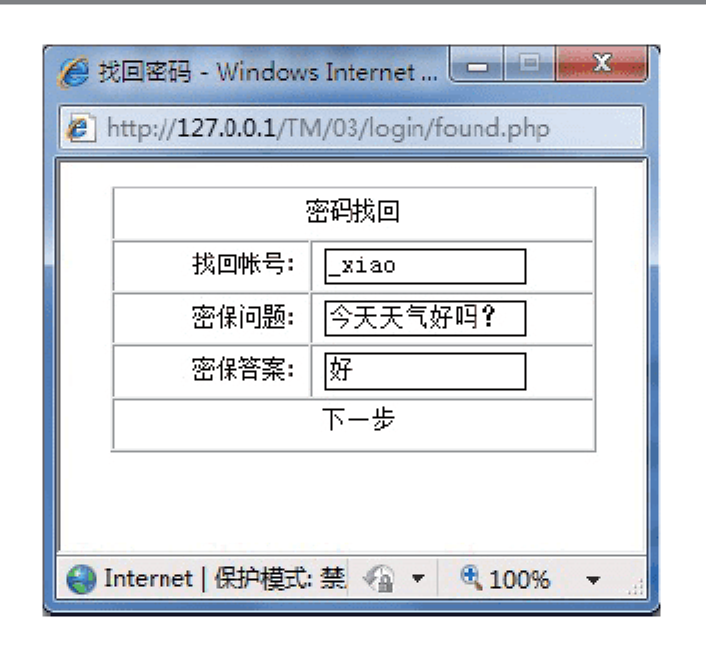
3.5.1 找回密码概述

长时间未登录的用户可能会忘记网站的登录密码；另外，一些用户的密码也可能会被他人非法更改。所以，找回密码模块即为个人用户取回密码提供了一个有效平台，这样就保证每一位用户能够及时找回登录密码。找回密码的运行结果如图 3.13 所示。

在本模块中，找回密码采用的是密保问题与密保答案比较的方式，首先用户输入注册的用户名，然后输入密保问题和密保答案，最后对用户的提交问题和答案与数据库中存储的数据进行比较，如果问题和答案都正确，则该用户可以执行密码的修改操作，重新设置新的密码。







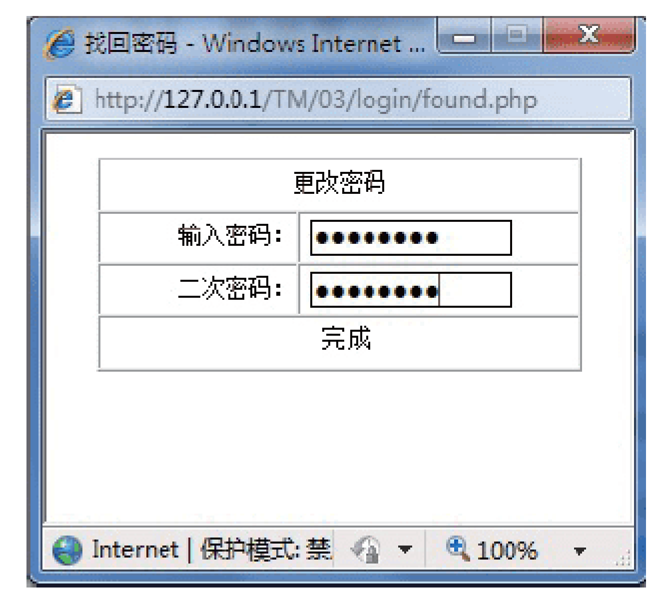


图 3.13　找回密码的运行结果

3.5.2　找回密码页面设计

在找回密码页面中，关键是创建表单，提交注册的用户名、密保问题和密保答案。同时还嵌入了两个不同 ID 值的 div 层，通过控制 div 层样式的隐藏和展开来实现找回密码表单和更改密码表单之间的切换操作。找回密码文件 found.php 的关键代码如下：

```
<!--找回密码部分-->
<div id="foundinfo">
<table width="250" border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="25" colspan="2" align="center" valign="middle">密码找回</td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">找回账号：</td>
<td height="25"> 
<input id="foundname" type="text" style=" width: 100px; height:15px; border:1px #000000 solid;" /></td>
</tr>
<!--省略部分代码-->
<tr>
<td height="25" colspan="2" align="center" valign="middle"> 
<button id="step1" class="btn">下一步</button></td>
</tr>
</table>
</div>
<!--更改密码部分-->
<div id="changepwd" style="display:none;">
<table width="250" border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="25" colspan="2" align="center" valign="middle">更改密码</td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">输入密码：</td>
<td height="25"> 
<input id="pwd1" type="password" style=" width: 100px; height:15px; border:1px #000000 solid;" /></td>
</tr>
<tr>
<td width="100" height="25" align="right" valign="middle">二次密码：</td>
```

```
<td height="25"> 
<input id="pwd2" type="password" style=" width: 100px; height:15px; border:1px #000000 solid;" /></td>
</tr>
<tr>
<td height="25" colspan="2" align="center" valign="middle"> 
<button id="step2" class="btn">完成</button></td>
</tr>
</table>
</div>
```

在 found.php 文件中，载入了 js\found.js 脚本文件，通过 Ajax 无刷新技术调用 found_chk.php 文件，对用户提交的密保问题和密保答案进行验证，并且根据 found_chk.php 文件中的返回值控制 div 层样式的隐藏和显示。found.js 脚本文件的代码如下：

```
$('step1').onclick = function(){
if($('foundname').value != '' && $('question').value != '' && $('answer').value != ''){
xmlhttp.open('get','found_chk.php?act=sel&foundname='+$('foundname')
.value+'&question='+$('question').value+'&answer='+$('answer').value,true);          //验证找回密码预设信息
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
msg = xmlhttp.responseText;
if(msg == '1'){
$('foundinfo').style.display = 'none';          //更改 display 属性隐藏 foundinfo 层
$('changepwd').style.display = '';              //更改 display 属性显示 changepwd 层
}else{
alert('填写信息错误！');
}
}
}
xmlhttp.send(null);
}
}
```

3.5.3　找回密码处理

创建找回密码处理文件 found_chk.php。该处理文件主要包括两个不同的处理方法，即验证密保问题和更改密码。首先通过 URL 获取 act 的参数值来判断用户执行的操作，然后根据选择的 act 参数来执行相应的方法，对用户提交的密保问题和密保答案进行验证，或者完成对用户提交新密码的更新操作。关键代码如下：

```
<?php
header('Content-Type:text/html;charset=gb2312');
include_once 'conn/conn.php';
$act = $_GET['act'];
$reback = '0';
if($act == 'sel'){                              //判断用户选择的操作
$name = $_GET['foundname'];                     //获取找回密码用户名
$question = $_GET['question'];                  //找回密码问题
```

```
$answer = $_GET['answer'];                          //找回密码
$sql="select * from tb_member where name='".$name."' and question='".$question."' and answer='".$answer."'";
$num = $conne->getRowsNum($sql);
}else if($act == 'up'){
$name = $_GET['foundname'];
$pwd = $_GET['pwd'];
$sql = "update tb_member set password = '".md5($pwd)."' where name = '".$name."'";
$num = $conne->uidRst($sql);                        //执行数据库更新操作
}
if($num != ''){
$reback = '1';                                      //返回值为 1
}else{
$reback = $sql;
}
echo $reback;
?>
```

说明 用户提交的新密码首先会经 MD5()函数加密，然后再被插入数据库中。

3.6　文件上传

3.6.1　文件上传概述

作为网络移动存储工具，上传是必不可少的功能之一。用户可以通过登录个人账户，单击“上传文件”超链接，进入文件上传页，将自己想要上传的工具、文档、音频或视频等多种类型的文件上传到个人空间中。文件上传页面的运行结果如图 3.14 所示。



图 3.14　文件上传页面的运行结果

3.6.2 文件上传表单设计

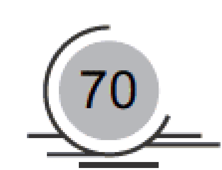
文件上传要通过表单中的文件域来提交，提交到处理页中通过$_FILES[]全局变量和 move_upload_file()函数完成文件的上传操作。

上传文件的表单存储于 upfile.php 文件中。首先，创建 form 标记，指定 POST 方法提交，设置 enctype 属性等于 multipart/form-data，设置处理页为 upfile_chk.php。然后，创建文件域，选择上传文件。接着，创建下拉列表框展示文件类型，将从数据库读取的文件类型作为下拉列表框的值，创建下拉列表框设置文件权限。最后，创建提交按钮，通过 onclick 事件调用 JavaScript 脚本中的方法验证提交数据，以及完成提交操作。upfile.php 文件的关键代码如下：

```
<!--上传表单，处理页为 upfile_chk.php-->
<form method="post" enctype="multipart/form-data" action="?act=upfile">
<!--列表项-->
<ul id="change">
<!--上传表单，注意，name 值为一个数组-->
<li id="addordel">
<input id="upname[]" name="upname[]" type="file" style=" width: 200px; background-color:#f0f0f0; " /> 
<!--文件类别下拉列表，name 也是数组-->
<select id="foundtype[]" name="foundtype[]">
<?php
$typesql = "select genrename from tb_uptype";              //查询语句
  $typearr = $conne->getRowsArray($typesql);               //返回查询结果集
$conne->close_rst();                                       //清空数组
foreach($typearr as $value){                               //循环输出类别列表
?>
<option value="<?php echo $value['genrename']; ?>"><?php echo $value['genrename']; ?></option>
<?php
}
?>
</select> 
<!--判断文件是否公开。0 表示不公开；1 表示公开-->
<select id="ispub[]" name="ispub[]">
<option value="0" selected="selected">不公开</option>
<option value="1">公开</option>
</select> 
<!--删除上传域和添加上传域-->
</li>
</ul>
<input id="upbtn" name="upbtn" type="submit" value="上传"
onClick="return chkfile()" style=" background-color:#f0f0f0; border: 1px #CCCCCC solid;" />
</form>
```

3.6.3 文件上传处理

创建文件上传处理页 upfile_chk.php。首先获取表单提交的数据，定义文件的存储路径。然后判断



是否存在重名文件，判断上传文件大小是否符合要求，生成下载码。接着完成文件的上传操作，并且将文件信息、下载码和空间使用信息存储到数据库中。最后输出下载码，返回上一页。关键代码如下：

```
include_once 'conn/conn.php';
$filename = $_FILES['upname']['name'];                  //读取上传文件名，并存为数组
$filetype = $_POST['foundtype'];                         //读取上传文件的类别
$tmpname = $_FILES['upname']['tmp_name'];               //读取临时文件名，并存为数组
$tmpsize = $_FILES['upname']['size'];                   //上传文件大小
$tmppub = $_POST['ispub'];                               //是否公开
$dirname=$_SESSION['name'];                              //选取和用户登录名相同的个人文件夹
$file_path = "upfile/$dirname/";                         //文件保存路径
$max = 0;                                                //文件大小，初始化为 0
for($j=0;$j<count($_FILES['upname']['name']);$j++) {    //判断上传的文件是否已存在
$filearr=$_FILES['upname']['name'][$j];                 //获取上传的文件名
$filenamei=$file_path.$filearr;                          //设置上传文件路径
if(file_exists($filenamei))                              //判断同目录下文件是否存在
$construct=0;
else
$construct=1;                                            //如果存在则返回 0，不存在则返回 1
}
if($construct!=1) {                                      //判断文件存在，则提示文件名称冲突
echo '<script>alert("文件名称冲突！请重新命名后上传！");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';
}
if(is_array($tmpsize)){                                  //如果变量$tmpsize 不是数组
foreach($tmpsize as $value){                             //获取文件总的大小
$max += $value;
}
if($max > 100000000 or $max <= 0){
echo '<script>alert("上传文件总大小大于 100MB，请重新选择");
location="'.$_SERVER['HTTP_REFERE R']. '";</script>';
}
}else{                                                   //重新选择上传文件
echo '<script>alert("上传文件错误，请重新检查程序");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';
}
$rightsql = "select * from tb_member where name = '".$_SESSION['name']."'";
$rightarr = $conne->getRowsRst($rightsql);
$conne->close_rst();
switch($rightarr['right']) {                             //判断用户权限
case '0':
$conmax=2000000;                                         //按当前登录用户的权限分配的最大使用空间
break;
case '1':
$conmax=3000000;
break;
case'2':
$conmax=5000000;
}
```

```
echo "<p>"
$lesarr=$conmax-$rightarr['content'];                    //将该用户剩余空间大小赋值给 lesarr 变量
if($conmax-$rightarr['content'] <= 1000000) {            //如果个人剩余空间已经不足 1MB 时
echo '<script>alert("您的上传下载空间不足，请清理后上传文件");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';   //提示空间不足
}
else if($lesarr-$max<=1000000)                           //如果上传文件大小超过剩余空间大小
{
echo '<script>alert("您的上传下载空间不足，请整理后上传文件");
location="'.$_SERVER['HTTP_REFERER'].'";</script>';    //提示空间不足
}
$chkdownnum = '';
for($i = 0; $i < 15; $i++){
$chkdownnum .= dechex(rand(0,15));
}
for ($i = 0; $i < count($filename); $i++){               //通过循环将多个文件全部上传
move_uploaded_file($tmpname[$i],$file—path.$filename[$i]);
//添加数据库
$insertsql = 'insert into tb_upfile (filename,filepath,filetype,upauthor,chkdownnum,ispub,filesize)
values("'.trim($filename[$i]).'","'.$file_path.$filename[$i].'","'.$filetype[$i].'",
"'.$_SESSION['name'].'","'.trim($chkdownnum).'",'.$tmppub[$i].','.$tmpsize[$i].')';
$conne->uidRst($insertsql);
}
$maxtotal=$rightarr['content']+$max;                     //从数据库得到该用户已使用空间大小与新上传的总数
//更新数据库
$csql="update tb_member set content = $maxtotal where name= '".$_SESSION['name']."'";
$conne->uidRst($csql);
echo '<div style=" padding-top: 30px;">文件上传成功，可以通过下载码：
<font color="#FF0000">'.$chkdownnum.'</font>  对文件进行下载。</div>';
echo "<p>点<a href=\"index.php?act=up\"><span style=\"color:red\">这里</span></a>继续上传其他文件！</p>";
```

说明 下载码是一组 16 位长的十六进制字符串。先由 rand()函数生成一个 0～15 之间的随机数，然后使用 dechex()函数将该数转为十六进制表示，最后使用 for 循环重复 16 次。如果一次上传多个文件，那么所有文件的注册码为同一个。

3.7 文 件 下 载

3.7.1 功能概述

文件下载是上传与下载模式的另一核心内容。首先，无论用户登录与否，都可以对已公开的文件进行下载；当用户登录后，不但可以上传新的文件，同时也可以对已经上传的文件执行下载操作和文件删除操作；如果当前用户没有登录，作为游客身份，也可以通过获得与文件匹配的下载码，对不公开文件执行下载操作。文件下载页的运行结果如图 3.15 所示。

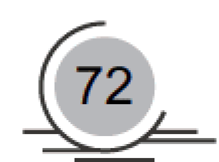




图 3.15　文件下载页的运行结果

3.7.2　通过下载码下载文件

用户登录后，可以通过“下载文件”超链接或输入下载码对文件进行下载。这里以下载码下载为例，讲解文件下载的实现过程。

下载码下载的操作在 index.php 文件中完成。首先，创建下载码文本框，提交下载码，其关键代码如下：

```
<div id="found"> 
下载码：
<input id="chkdown" type="text" /> 
<button id="downbtn" onclick="chkdown()"></button>
```

在 index.php 文件中，当用户单击“下载”按钮时，将触发 onclick 事件，调用 chkdown()函数判断输入值是否为空，如果不为空，将重新加载本页，并将下载码作为参数传递。chkdown()函数存储于 js\index.js 脚本文件中，其关键代码如下：

```
function chkdown(){
if($('chkdown').value == ''){
alert('请输入下载码!');
return false;
}
location = '?act=downcode&chkdown='+$('chkdown').value;
}
```

然后，在 index.php 页中，当超链接传递的值等于 downcode 时，加载 downcode.php 页。

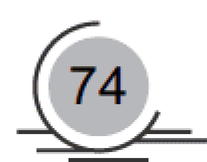
```
<!--根据不同的参数值，载入不同的页面-->
<div id="content">
```



```
<?php
$act = $_GET['act'];
switch($act){
case 'up':
include 'upfile.php';                              //载入上传页
break;
case 'upfile':
include 'upfile_chk.php';                          //载入上传处理页
break;
case 'down':
  include 'downfile.php';                          //载入下载页
  break;
case 'downcode':
include 'downcode.php';                            //载入下载码下载页
break;
default:
include 'pub.php';                                 //载入共享文件页
  break;
}
?>
</div>
```

接着，创建 downcode.php 页，根据获取的下载码生成 SQL 查询语句，循环输出下载文件信息，并且创建文件下载超链接，链接到 download.php 文件，完成文件下载操作。关键代码如下：

```
<?php
$chkdown = trim($_GET['chkdown']);
$downsql = "select * from tb_upfile where chkdownnum = '".$chkdown."'";
$downarr = $conne->getRowsArray($downsql);
?>
<!--下载文件列表-->
<div id="filelist">
<?php
$infoline = 0;
if(!empty($downarr)){                              //如果结果集数组不为空
?>
<ul>
  <li style=" width: 30px; background-color: #F8E7A5;"> </li>
<li style=" width: 200px; background-color:#F8E7A5; ">文件名</li>
<li style=" width: 50px; background-color:#F8E7A5; ">文件类型</li>
  <li style=" width: 100px; background-color:#F8E7A5;">下载码</li>
  <li style=" width: 150px; background-color:#F8E7A5;">上传时间</li>
</ul>
<?php
foreach($downarr as $key => $value){               //循环输出数组
?>
<ul>
<li style=" width: 30px;background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
<input id="chk[<?php echo $key; ?>]" type="checkbox" value="<?php echo $value['id']; ?>"
style=" width:30px; height:20px; border: 0px;" /></li>
```



```
    <li style=" width: 200px; background-color:<?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <a href="download.php?id=<?php echo $value['id']; ?>"><?php echo $value['filename']; ?></a></li>
    <li style=" width: 50px; background-color:   <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['filetype']; ?></li>
<li style=" width: 100px; background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['chkdownnum']; ?></li>
    <li style=" width: 150px; background-color: <?php echo ($infoline %2 ==0?'#FFFFFF':'#f0f0f0'); ?>;">
  <?php echo $value['uptime']; ?></li>
</ul>
<?php
$infoline = ($infoline+1)%2;
}
}else{
echo '没有可下载资源';
}
?>
</div>
```

最后，创建 download.php 文件，根据文件的完整路径和 header()函数完成下载。关键代码如下：

```
<?php
include_once 'conn/conn.php';
$id = $_GET['id'];                                      //获取用户的 id
$downsql = "select * from tb_upfile where id = $id";    //通过 where 子句获得相应的文件
$downarr = $conne->getRowsRst($downsql);
$path=$downarr['filepath'];                             //获取路径名称赋给$path
if(!empty($path) and !is_null($path)){
$filename=basename($path);
$file=fopen($path,"r");
header("Content-type:application/octet-stream");
    header("Accept-ranges:bytes");
header("Accept-length:".filesize($path));
header("Content-Disposition:attachment;filename=".$filename);
echo fread($file,filesize($path));
fclose($file);
    exit;
}
?>
```

3.7.3　同步删除文件以及释放用户空间

单击“下载文件”超链接后会在下载文件的下方看到删除选择的文件，在删除数据库文件的同时，也会删除该登录用户相应文件夹下的同名文件，并且同步更新用户使用空间的大小，达到删除文件释放空间的效果，删除文件后，1 秒钟之后返回下载首页。删除文件释放空间的操作在 downfile.php 文件中执行，其关键代码如下：

```
<?php
$del = $_GET['del'];
$num =10;                                   //每页显示记录数
```

```
$curpage = $_GET['curpage'];                    //当前页
$sql = "select * from tb_upfile where upauthor = '".$_SESSION['name']."'";
$totnum = $conne->getRowsNum($sql);                                   //记录数
if($del == 'del'){
$rd = $_GET['rd'];
$delsql = "delete from tb_upfile where id = -1 ";
$deltotal = "select sum(filesize) as total from tb_upfile where id = -1 ";        //取得选择文件的总大小
$content="select * from tb_member where name ='".$_SESSION['name']."'";
$contarr = $conne->getRowsRst($content);
if(!empty($rd)){
$delarr = explode(',',$rd);
foreach($delarr as $value){
//以下为新加内容，在删除文件的同时也可以删除硬盘文件夹上的文件
$sql= "select * from tb_upfile where id = ".$value." ";        //选取删除文件的 id 值
$delsql .= " or id = ".$value." ";
$deltotal .= " or id = ".$value." ";
$sqlarr = $conne->getRowsRst($sql);                        //调用 SQL 函数
$delpath=$sqlarr['filepath'];                              //获得删除文件路径
unlink($delpath);                                          //删除指定路径的文件
}
$deltotalnum=$conne->getRowsRst($deltotal);
$newdata = $contarr['content']-$deltotalnum['total'];
// $newdata 为删除文件后个人空间的使用大小
$newsql="update tb_member set content = $newdata where name = '".$_SESSION['name']."'";
$conne->uidRst($newsql);                                   //把删除文件后剩余空间的大小更新到数据库
$delnum = $conne->uidRst($delsql);
echo "<meta http-equiv=\"refresh\" content=\"1;url=index.php?act=down\">";    //1 秒钟之后跳转到主页
echo "<p></p>";
echo "删除成功！ ";
}
```

3.8 后台管理

3.8.1 后台管理概述

在后台管理系统中只有一个功能，就是管理注册用户的权限。其运行结果如图 3.16 所示。

用户权限管理

目前共有2条记录　共分1页显示　当前显示第1页

用户名	使用权限	使用容量	修改
mrsoft	高级用户	0.00MB	修改
leonsk	高级用户	0.00MB	修改

第一页 | 上一页 | 下一页 | 最后一页

图 3.16　用户权限管理

3.8.2 用户权限管理页面设计

进入到后台管理系统中，以分页形式输出已经注册的用户信息，包括用户名、使用权限和使用容量，以及“修改”超链接。admin.php 的关键代码如下：

```
<?php
include_once 'conn/conn.php';
```

```
$list_num=10;
$sql = "select * from tb_member";                              //查询语句
$sqlarr = $conne->getRowsRst($sql);                            //返回查询结果集
$totnum = $conne->getRowsNum($sql);                            //获取查询记录数
$conne->close_rst();
if($totnum>0){
?>
<p>
<table>
<tr class="tr1">
<td>用户名</td>
<td>使用权限</td>
<td>使用容量</td>
<td>修改</td>
</tr>
<?php
$sql = $sql." order by id desc limit ".($list_num *($page-1)).",".$list_num;  //查询语句
$filearray = $conne->getRowsArray($sql);
foreach($filearray as $value){
echo "<tr>\n";
echo "<td>".$value['name']."</td>\n";
switch($value['right']){                                       //查看用户权限
case '0':
echo "<td>普通用户</td>\n";
break;
case '1':
echo "<td>会员用户</td>\n";
break;
case '2':
echo "<td>高级用户</td>\n";
break;
}
echo "<td>".sprintf("%-01.2f",($value['content']/1000000))."MB</td>\n";
echo "<td><a href=changesub.php?id=".$value['id']."">修改</a></td>\n";   //获取 id 更改用户权限
echo "</tr>\n";
}
echo "</table>";
```

3.8.3　更改用户权限表单设计

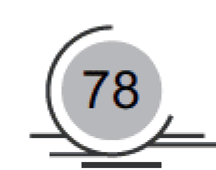
更改用户权限从管理页中设置的“修改”超链接开始，在 changesub.php 文件中，根据超链接传递的 ID 值，查询出指定用户的详细信息，并且将该信息在表单中输出，进而通过表单将数据提交到指定页完成对用户权限的修改操作。changesub.php 文件的关键代码如下：

```
<table border=1>
<form action="change.php" method="post">
<?php
include_once 'conn/conn.php';
```

```
$sql="select * from tb_member where id = $id";                    //查询语句
$changearr = $conne->getRowsRst($sql);                            //返回查询结果
?>
<tr>
<td>用户名</td>
<td><?php echo $changearr['name'];?></td>
</tr>
<tr>
<td>当前权限</td>
<td>
<?php
switch($changearr['right']){                                      //获取用户权限
case '0':
echo "普通用户";
break;
case '1':
echo "会员用户";
break;
case '2':
echo "高级用户";
break;
}
?>
</td>
</tr>
<tr>
<td>修改权限</td>
<td>
<select name="right">                                             //选择用户权限
<option value="0" selected="selected">普通用户</option>
<option value="1">会员用户</option>
<option value="2">高级用户</option>
</select>
</td>
</tr>
<tr><td colspan=2>
<center>
<input type="submit" value="提交" />
</center>
</td>
</tr>
<input type="hidden" name="id" value="<?php echo $changearr['id']?>" />//添加一个隐藏域传递 id 参数
</form>
</table>
```

3.8.4 执行权限更改操作

创建在 changesub.php 中指定的权限更改处理文件 change.php，用于获取表单中提交的数据，完成用户权限的更改操作。change.php 文件的关键代码如下：



```
<?php
include_once 'conn/conn.php';
$right=$_POST['right'];                                        //获得权限更改表单的传递值
$id=$_POST['id'];
$upsql = "update tb_member set tb_member.right =$right where id = $id";    //查询语句
$num=$conne->uidRst($upsql);                                   //返回结果
$conne->close_rst();
echo "<p>";
if($num==1){
echo "权限更改成功！";
echo "<p>";
}else{
echo "权限更改失败！";
echo "<p>";
}
?>
<!--以下为 html 内容，修改成功会返回修改主页-->
<meta http-equiv="refresh" content="2;url=admin.php" />
2seconds   later to return the manger page!
```

3.9　技 术 提 炼

3.9.1　php.ini 配置文件

如果要在 PHP 中实现小文件的上传（2MB 以下），那么无须对 php.ini 配置文件进行修改，使用默认参数即可。但如果想实现完美的上传功能，则一定要对配置文件有所了解。在 php.ini 文件中，与文件上传有关的设置如图 3.17 所示。

```

;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;

; Maximum execution time of each script, in seconds
max_execution_time = -1
; Maximum amount of time each script may spend parsing request data
max_input_time = -1
; Maximum amount of memory a script may consume (16MB)
memory_limit = 32M


; Maximum size of POST data that PHP will accept.
post_max_size = 16M


;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system defau
; specified).
upload_tmp_dir = "D:\xampp\tmp"

; Maximum allowed size for uploaded files.
upload_max_filesize = 32M
```

图 3.17　php.ini 文件中的相关参数

从图 3.17 中可以看到，有 3 部分内容需要进行配置，分别介绍如下。

1. Resource Limits

Resource Limits，直译过来就是资源限制，包含了 3 个参数。该区块不仅仅是针对上传与下载的，而是对全部的文件进行设置。各个参数含义及参数值说明如表 3.1 所示。

表 3.1 Resource Limits 块的参数说明

参　数	描　述
max_execution_time	每个脚本页面完成执行操作的最大时间，单位是秒。如果设为-1，说明没有限制
max_input_time	每个脚本页面处理请求数据的最大时间，单位是秒，也可以设为-1
memory_limit	一个脚本页所能够消耗的最大内存

2. post_max_size

post_max_size 参数指 PHP 通过表单 POST 所能接收的最大值，包括表单中的所有项。

3. File Uploads

File Uploads 块是专为文件上传设置的。该区块中同样包含了 3 个参数。参数含义及参数值说明如表 3.2 所示。

表 3.2 File Uploads 块的参数说明

参　数	描　述
file_uploads	是否允许 HTTP 上传，默认为 On，即为开启，无须修改
upload_tmp_dir	文件上传时的临时存储目录。如果没指定就会用系统默认的临时文件夹
upload_max_filesize	允许上传文件的最大值

3.9.2 上传表单设置

配置文件修改完毕后，重启 Apache 服务器，接下来开始设计<form>表单。<form>表单可以实现人机交互，是动态网页的一个最基本功能。最初的<form>表单不支持文件上传，表单中 input 元素的 type 属性只有 7 种值：文本框（text）、密码框（text）、单选按钮（radio）、复选框（checkbox）、重置按钮（reset）、确定按钮（submit）和图像（image）。后来，rfc1867 为 HTTP 协议添加了这个功能，目前几乎所有的浏览器和服务器都支持此协议。

rfc1867 协议为 type 属性增加了 file 值，同时，<form>标签中的 enctype 属性值必须设为 multipart/form-data，method 属性则必须是 POST，3 个条件缺一不可。当表单属性设置正确后，才可以在表单处理页中获取到上传文件的值。下面是简单、可用的上传表单代码。

```
<form method="post" action="2.php" enctype="multipart/form-data">
        <input name="filename" type="file" />
        <input type="submit" value="enter"/>
</form>
```

此外，还建议在上传表单中增加一个 hidden 隐藏域，隐藏域的 name 值为 MAX_FILE_SIZE，value



值是允许上传的最大字节数。例如：

```
<input type="hidden" name="MAX_FILE_SIZE" value="1024000" />
```

这样，当上传文件大于 1MB 时，$_FILES 中的 error 值将返回 2。关于$_FILES 变量的使用在 3.9.3 节中给出。该隐藏域的作用不是真正要去限制上传文件的大小，因为要绕过该项检查非常简单，它的真正作用是为了避免用户误传了大文件而陷入“无尽的”等待之中。


注意 通过隐藏域控制上传文件大小，必须将隐藏域放置在文件域之前，否则隐藏域不会生效。

3.9.3　预定义变量$_FILES

如果表单设计得没有问题，那么在处理页中，使用$_FILES 预定义变量就可以得到上传文件的相关信息；反之，如果得不到，那么就有可能是配置文件或表单设置出问题了。下面先介绍预定义变量$_FILES。

$_FILES 变量存储的是上传文件的相关信息，这些信息对于上传功能有很大作用。该变量是一个二维数组，保存的信息如表 3.3 所示。

表 3.3　$_FILES 变量值

参　数	描　述
$_FILES[filename]['name']	存储了上传文件的文件名。例如 exam.txt、myDream.jpg 等
$_FILES[filename]['size']	存储了文件大小。单位为字节
$_FILES[filename]['tmp_name']	文件上传时，首先在临时目录中被保存成一个临时文件。该变量为文件名
$_FILES[filename]['type']	上传文件的类型。注意是类型，不是后缀
$_FILES[filename]['error']	存储了上传文件的错误代码

error 字段将返回上传文件的错误码，一共有 7 种状态。错误代码及说明如表 3.4 所示。

表 3.4　$_FILES[filename]['eror']中的错误代码及说明

错误代码	错误说明
0	UPLOAD_ERR_OK。上传成功，无错误
1	UPLOAD_ERR_INI_SIZE。文件大小超出 php.ini 中 upload_max_filesize 的参数值
2	UPLOAD_ERR_FORM_SIZE。文件大小超出表单中 MAX_FILE_SIZE 隐藏域的值
3	UPLOAD_ERR_PARTIAL。文件没有被全部上传
4	UPLOAD_ERR_NO_FILE。没有文件被上传
6	UPLOAD_ERR_NO_TMP_DIR。找不到临时文件目录
7	UPLOAD_ERR_CANT_WRITE。文件写入失败

下面使用 3.6.2 节中的表单上传一个文件，使用$_FILES 变量输出文件信息。代码如下：

```
<?php
$tmparr = $_FILES['filename'];                    //获取文件域的值
if($tmparr['name'] != ''){                        //如果上传文件名不为空
```

```
foreach($tmparr as $name => $value){
echo $name.' => '.$value.'<br>';                    //使用循环输出数组索引和值
}
}else{
echo "<script>alert('请添加上传文件');history.go(-1);</script>";
}
?>
```

运行结果如图 3.18 所示。

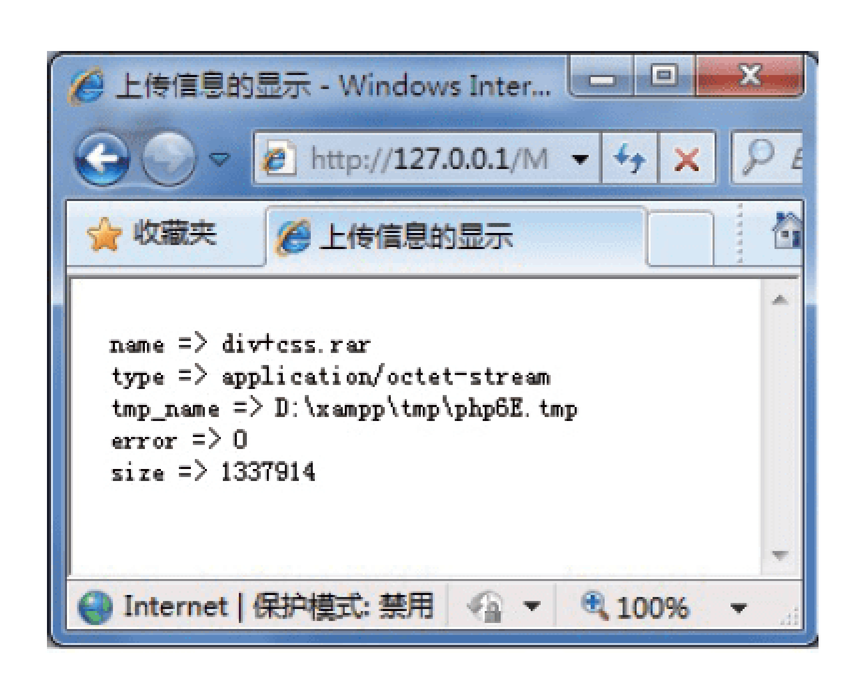


图 3.18　获取上传文件的信息

3.9.4　上传文件函数

move_uploaded_file()函数用来上传文件到指定的位置。如果成功返回 true，否则返回 false。该函数的语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

☑　filename：上传文件的临时文件名，即$_FILES['tmp_name']。
☑　destination：上传后保存的新的路径和名称。

通过 3.6.3 节和 3.7.3 节的阅读，细心的读者可能已经发现，真正的上传文件已经被保存到临时目录中，move_uploaded_file()函数的作用只是将文件换个地方保存，这是出于安全的考虑。

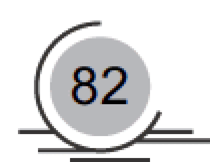
此外，还可以通过 is_uploaded_file()函数判断文件是否为 POST 上传。该函数的语法如下：

```
bool is_uploaded_file ( string filename )
```

3.9.5　多文件上传

多文件上传和单文件上传没什么区别，只要将表单中的文件域用数组命名即可。代码如下：

```
<form action="2.php" method="post" enctype="multipart/form-data">
<input name="userfile[]" type="file" /><br />
<input name="userfile[]" type="file" /><br />
…
<input type="submit" value="上传" />
</form>
```





上传的多文件可以通过 for 循环语句进行获取。方法如下：

```
$filename= $_FILES['userfile']['name'];
for ($i = 0; $i < count($filename); $i++){              //通过循环将多个文件全部上传
move_uploaded_file($tmpname[$i],$file_path.$filename[$i]);
//添加数据库
$insertsql = 'insert into tb_upfile (filename,filepath,filetype,upauthor,chkdownnum,ispub)
values("'.trim($filename[$i]).'","'.$file_path.$filename[$i].'","'.$filetype[$i].'",
"'.$_SESSION['name'].'","'.trim($chkdownnum).'",'.$tmppub[$i].')';
$conne->uidRst($insertsql);
}
echo '<div style=" padding-top: 30px;">文件上传成功
<font color="#FF0000">'.$chkdownnum.'</font> 对文件进行下载。</div>';
```

说明 本模块中并没有加入多文件上传功能，因为多文件上传存在一些弊端，如在上传多个文件时，多个文件会获取相同的下载码。在上传多个文件时，判断多个文件上传是否有重名比较麻烦，一旦上传失败，会导致所有文件都上传失败，必须重新操作上传步骤。这时可以在上传页面增加 JavaScript 脚本，在提交之前判断是否存在同名文件，如果不加入这个判断过程，在其后删除多个同名文件时就会导致出错。有兴趣的读者可以自行研究一下。

3.9.6　文件下载

在一个完整的网站中，拥有文件下载功能能够给网站带来更多的访问者，增加网站的访问量。实现下载最常用的方式有通过链接下载和通过 header()函数下载。

（1）通过链接下载

通过链接下载，就是将文件的相对路径或绝对路径直接作为超链接即可。如果是压缩文件，那么直接单击超链接就可以下载；对于文本文件，单击鼠标右键，在弹出的快捷菜单中选择“目标另存为”命令，就可以将文本文件下载下来。所有格式的文件都可以通过这种方法下载，运行结果如图 3.19 所示。

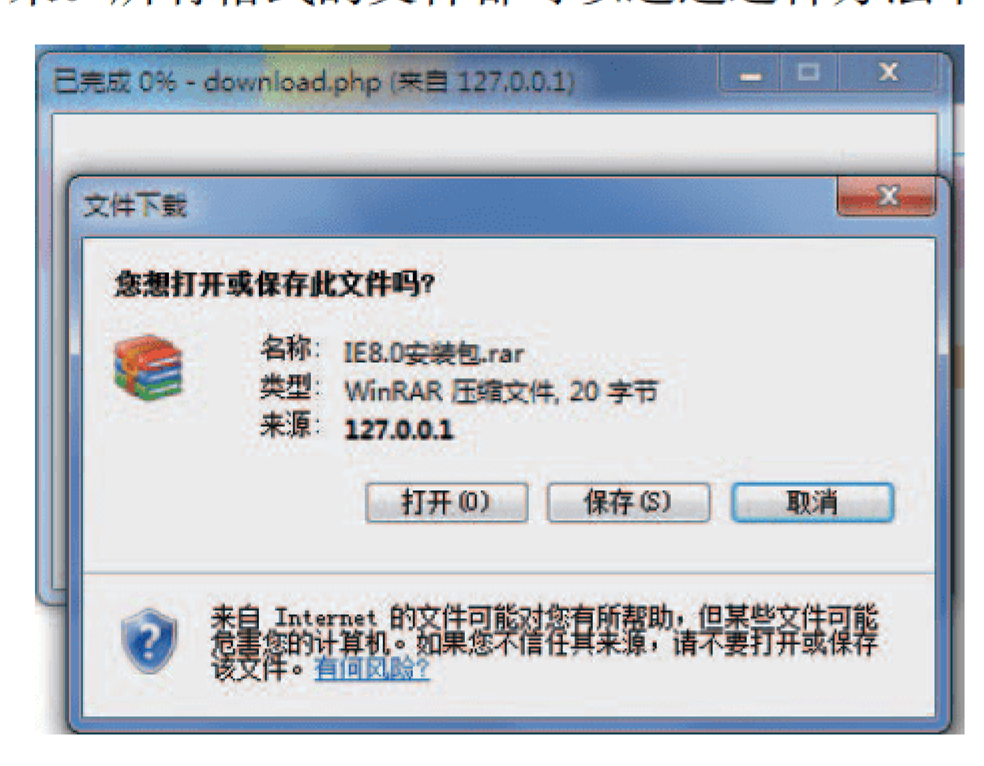


图 3.19　通过链接下载

（2）通过 header()函数下载

header()函数属于 HTTP 函数，它的作用是发送一个原始的 HTTP 标头。header()函数的语法如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

- ☑ string：发送的标头。
- ☑ replace：如果一次发送多个标头，指明相似的标头是替换还是添加。如果是 false，则强制发送多个同类型标头。默认是 true，即替换。
- ☑ http_response_code：强制 HTTP 响应为指定值。

HTTP 标头有很多，这里只介绍下载，其他请参考相关资料。下载 HTTP 标头的代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

上面的代码中唯一需要改动的就是 filename。将 filename 替换为要下载的文件就可以完成下载。

3.10 本章总结

本章通过详细讲解一个完整的上传与下载模块。通过该模块可以将要上传的工具、文档、音频或视频等多种类型的文件上传到个人空间中。文件下载是上传与下载的另一核心内容。首先，无论用户登录与否，都可以对已公开的文件进行下载；当用户登录后，不但可以上传新的文件，同时也可以对已经上传的文件执行下载操作和文件删除操作。

第 4 章

FTP 管理系统

（PHP+Session 实现）

FTP 管理工具作为一种服务器管理系统，其为管理者通过互联网执行远程操作或管理服务器带来了极大方便。FTP 协议是 Internet 上的文件传输通用协议，它位于 TCP/IP 协议的应用层。利用该协议，用户可以将一个完整的文件从一台计算机复制到另一台计算机上。但是在使用 FTP 传输文件之前，需要登录 FTP 服务器，用户可以通过已设置的 FTP 用户名和密码登录服务器。本章向读者展现一个多功能 FTP 管理系统，主要实现本地系统与远程 FTP 间文件的上传下载和管理文件等功能。

通过阅读本章，可以学习到：

- 遍历 FTP 服务器目录
- 获取本地文件的图标
- 实现 FTP 文件上传下载
- 本地文件与 FTP 文件的维护操作

4.1 FTP 管理系统概述

4.1.1 模块概述

FTP 管理系统是一个可以执行常用操作的上传下载工具，其作用是方便用户快捷地访问 FTP 服务器，应用 FTP 传输文件的独立性。用户与用户在分别执行各自的上传、下载和维护等操作时互不影响。另外，FTP 管理系统具有良好的远程管理性，用户可以通过互联网登录 FTP 服务器，执行远程管理操作。

4.1.2 功能结构

FTP 管理系统包括根目录操作和子目录操作两大基本结构。其中根目录操作包括查看文件、查看目录、删除目录、创建目录和上传文件；子目录操作包括查看文件、查看目录、更改文件、移动文件、删除文件、创建目录和上传文件。FTP 管理系统的功能结构如图 4.1 所示。

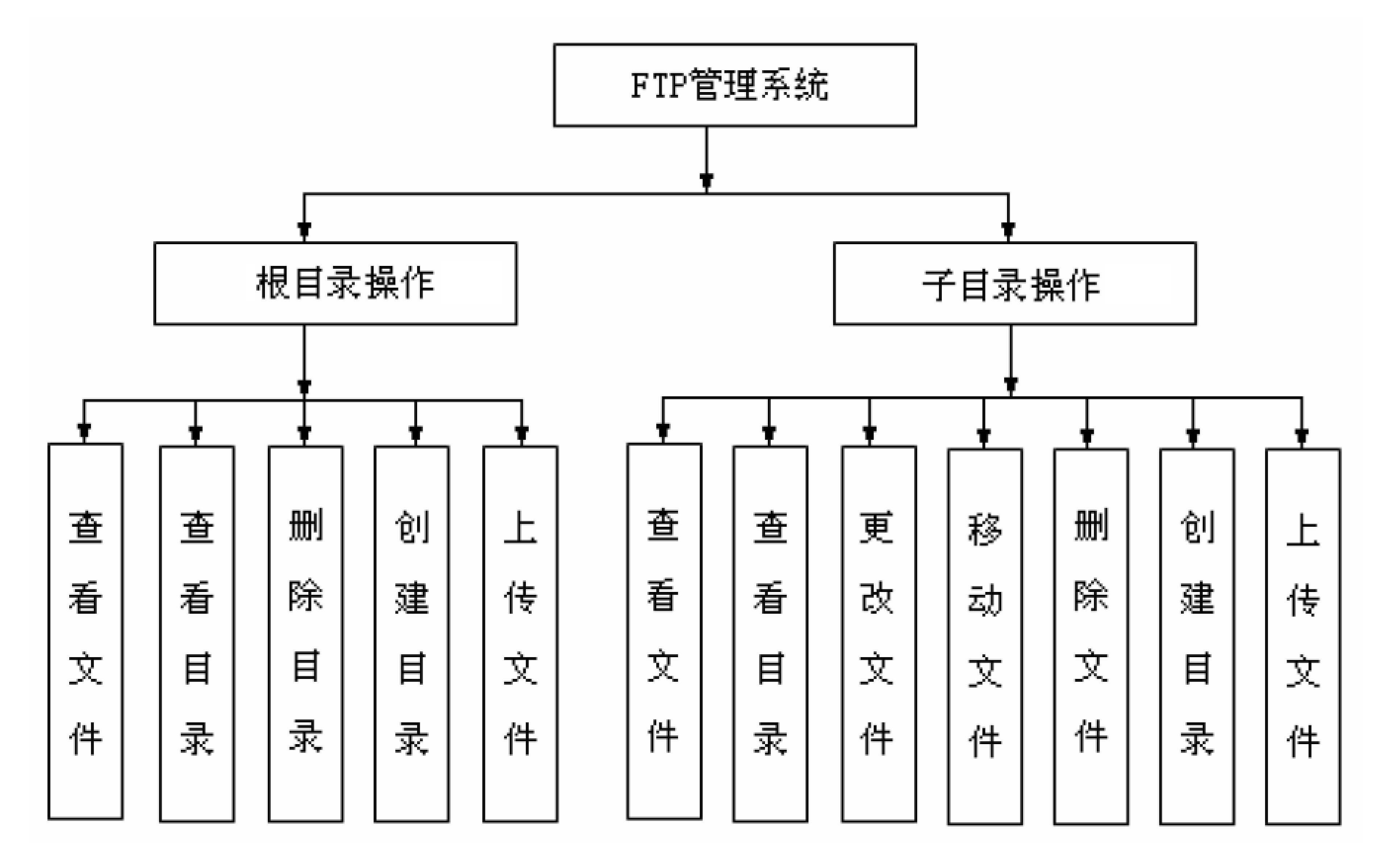


图 4.1　FTP 管理系统的功能结构

4.1.3 程序预览

为了让读者对本系统有个初步的了解和认识，下面给出本系统的几个页面运行效果图，如果想查看完整的效果图，请参见光盘源程序。

FTP 管理系统的登录页面和主页面的运行效果如图 4.2 和图 4.3 所示。

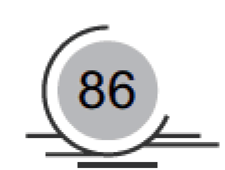




图 4.2　登录页面的运行效果



图 4.3　FTP 主页面的运行效果

子目录下的文件详细信息运行结果如图 4.4 所示。



图 4.4　子目录下的文件详细信息运行结果

单击文件名称的超链接，下载 FTP 文件的运行结果如图 4.5 所示。单击“移动”超链接，移动文件到指定文件目录的操作如图 4.6 所示。

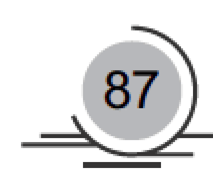




图 4.5　下载 FTP 文件的运行结果



图 4.6　移动文件到指定文件目录的操作

4.2　FTP 环境搭建

4.2.1　FTP 服务器端概述

FTP 服务器端是一种可以将计算机或服务器设置成 FTP 服务器的软件工具。通过 FTP 软件，用户或其他使用者就能够使用 FTP 协议，通过互联网或者同一网络上的任何一台计算机与 FTP 服务器连接，执行文件或目录的复制、移动、创建和删除等操作。这里提到的 FTP 协议是专门被用来规定计算机之间进行文件传输的标准和规则，正是因为有了像 FTP 这样的专门协议，才使得人们能够通过不同类型的计算机，使用不同类型的操作系统，实现不同类型的文件进行相互传递。

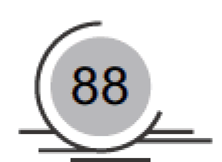
目前 FTP 软件的种类繁多，其功能各有千秋。其中 Serv-U 是应用最为广泛的 FTP 软件之一，其凭借优秀的功能博得人们的青睐。Serv-U 软件可以提供以下功能：

- ☑ 符合 Windows 标准的用户界面友好亲切，易于掌握。
- ☑ 通过限制同一时间最大的用户访问人数确保 PC 的正常运转。
- ☑ 安全性能出众。在目录和文件层次中都可以设置安全防范措施。
- ☑ 能够为不同用户提供不同设置，支持分组管理数量众多的用户。
- ☑ 可以基于 IP 对用户授予或拒绝访问权限。
- ☑ 支持文件上传和下载过程中的断点续传。
- ☑ 支持拥有多个 IP 地址的多宿主站点。
- ☑ 能够设置上传和下载的比率、硬盘空间配额、网络使用带宽等，从而能够保证用户有限的资源不被大量的 FTP 访问用户所消耗。
- ☑ 可作为系统服务后台运行。
- ☑ 可自行设置在用户登录或退出时的显示信息，支持具有 UNIX 风格的外部链接。

4.2.2　Serv-U 的安装和配置

为了保证 FTP 管理系统能够正常运行并发挥其应有的功能效果，在使用 FTP 管理系统前应在远程服务器上搭建 FTP 服务环境。本系统选择 Serv-U 来搭建 FTP 服务环境。FTP 服务环境的架设过程如下：

（1）用户可以通过互联网下载 Serv-U FTP Server 安装包。将下载好的安装解压缩到指定文件目





录中，并双击 ServUSetup.exe 打开安装文件。

（2）选择语言为中文（简体）形式，并单击“确定”按钮进入下一步安装向导设置。安装向导运行界面如图 4.7 所示。

（3）单击“下一步”按钮，进入许可协议查看界面。选择“我接受协议”后，继续单击“下一步”按钮，进入选择目标位置界面，如图 4.8 所示。



图 4.7　安装向导运行界面



图 4.8　选择目标位置界面

（4）设置路径完毕后，继续单击“下一步”按钮，进入准备安装界面，如图 4.9 所示。单击“安装”按钮开始安装软件。

（5）当软件的所有部分安装完毕后，弹出如图 4.10 所示的界面。

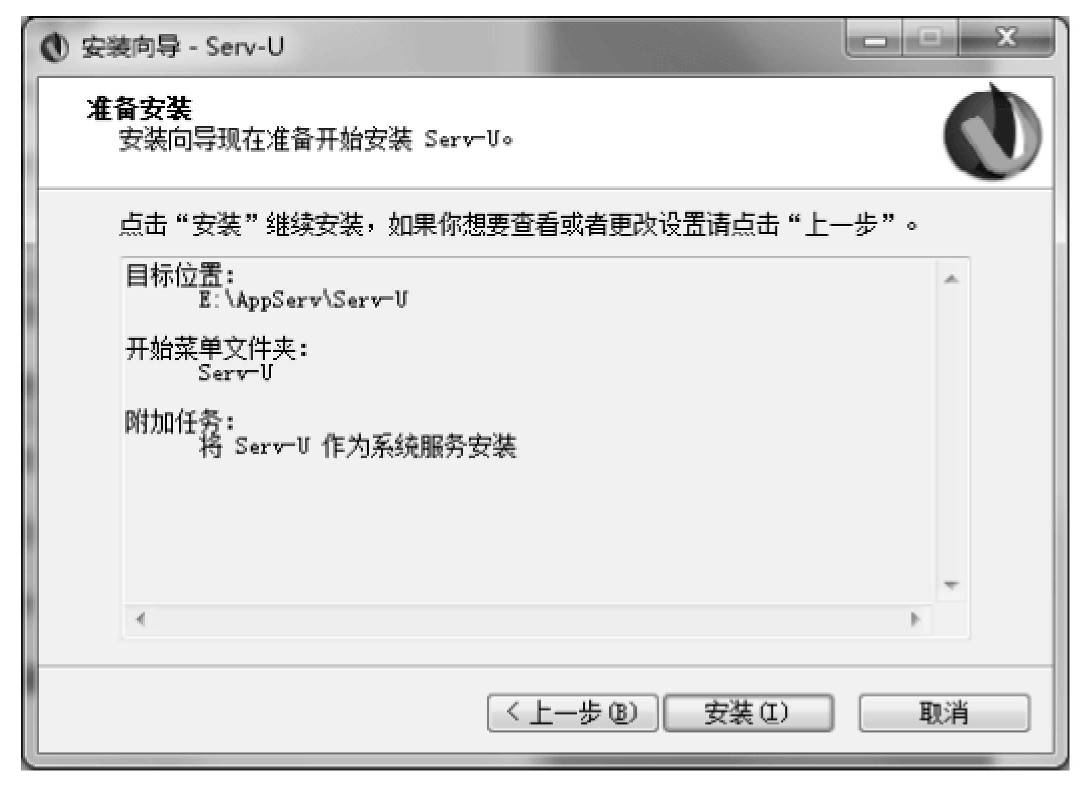



图 4.9　准备安装界面



图 4.10　安装完成

这样 FTP 服务器端软件 Serv-U 即安装完毕。用户单击“完成”按钮来运行 Serv-U 软件，系统会提示用户设置相应的必要参数。下面讲解如何配置 Serv-U。

（1）定位到新添加的文件目录，找到可执行文件 Serv-U-Tray.exe，双击该程序即可启动本地 FTP 服务端，进入控制管理台，首先需要设置上传。其中管理控制台的运行界面如图 4.11 所示。

（2）单击“域详细信息”选项，进入域创建界面。如果为首次设置域信息，则需要新建域，如图 4.12 所示。

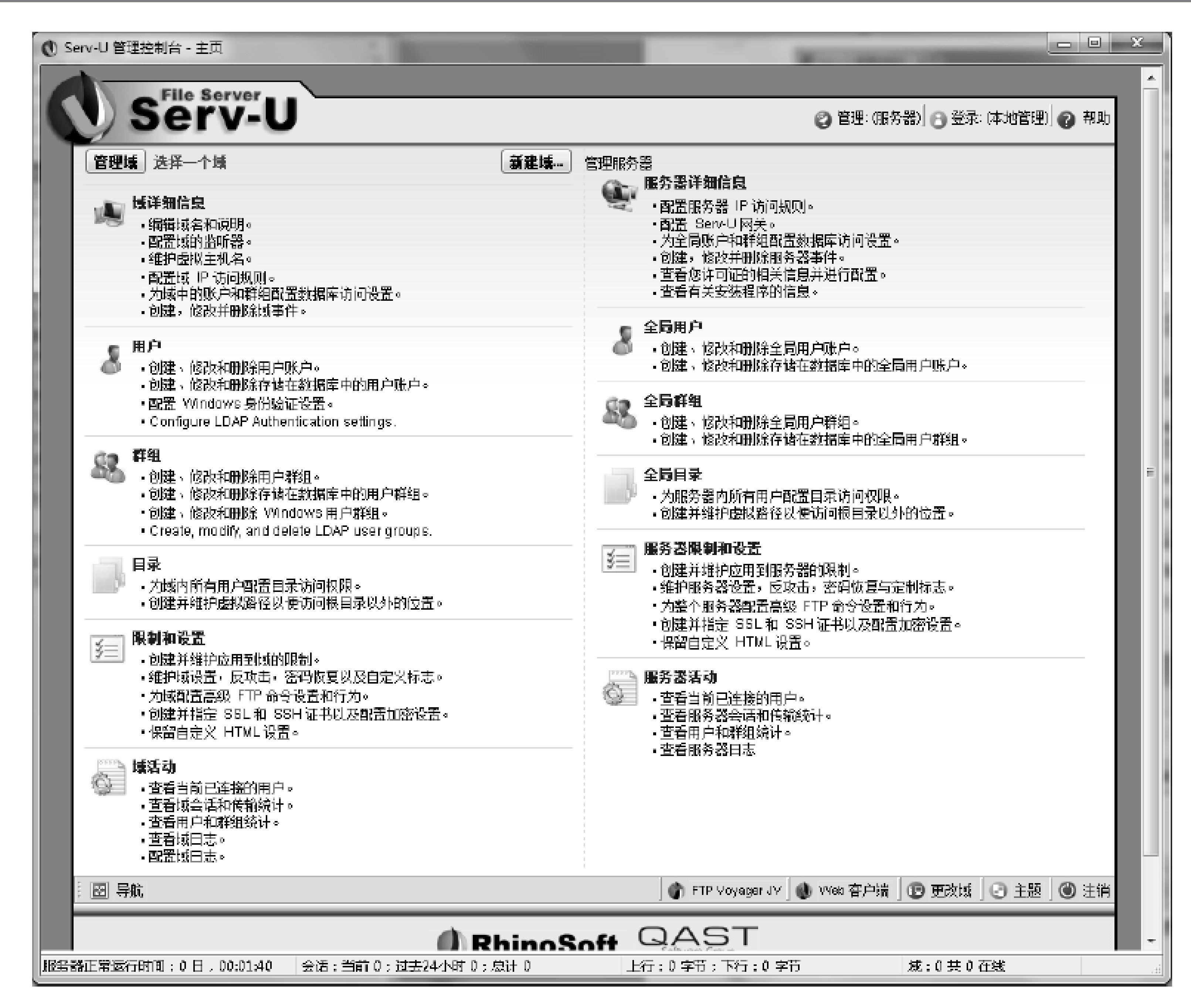


图 4.11　管理控制台的运行界面

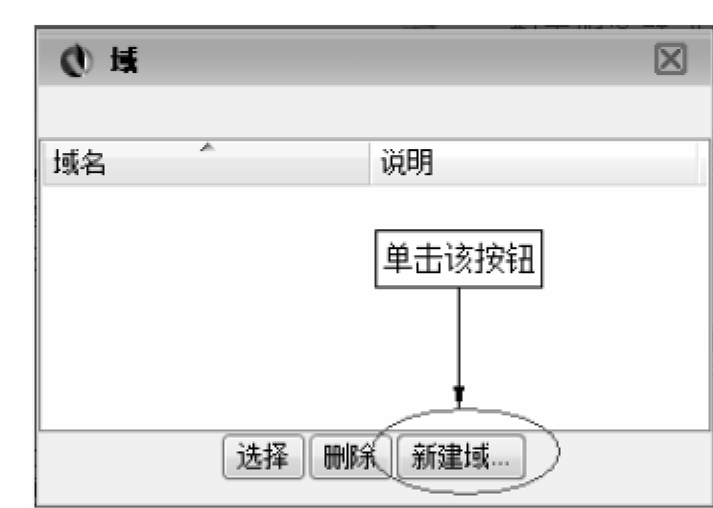


图 4.12　单击“新建域”按钮设置域信息

（3）进入“域向导”设置界面。需要进行 4 步设置：① 域名称；② FTP 协议、端口；③ 可用 IP 地址；④ 密码加密方式。其中，设置 FTP 端口、密码加密方式的界面如图 4.13 和图 4.14 所示。

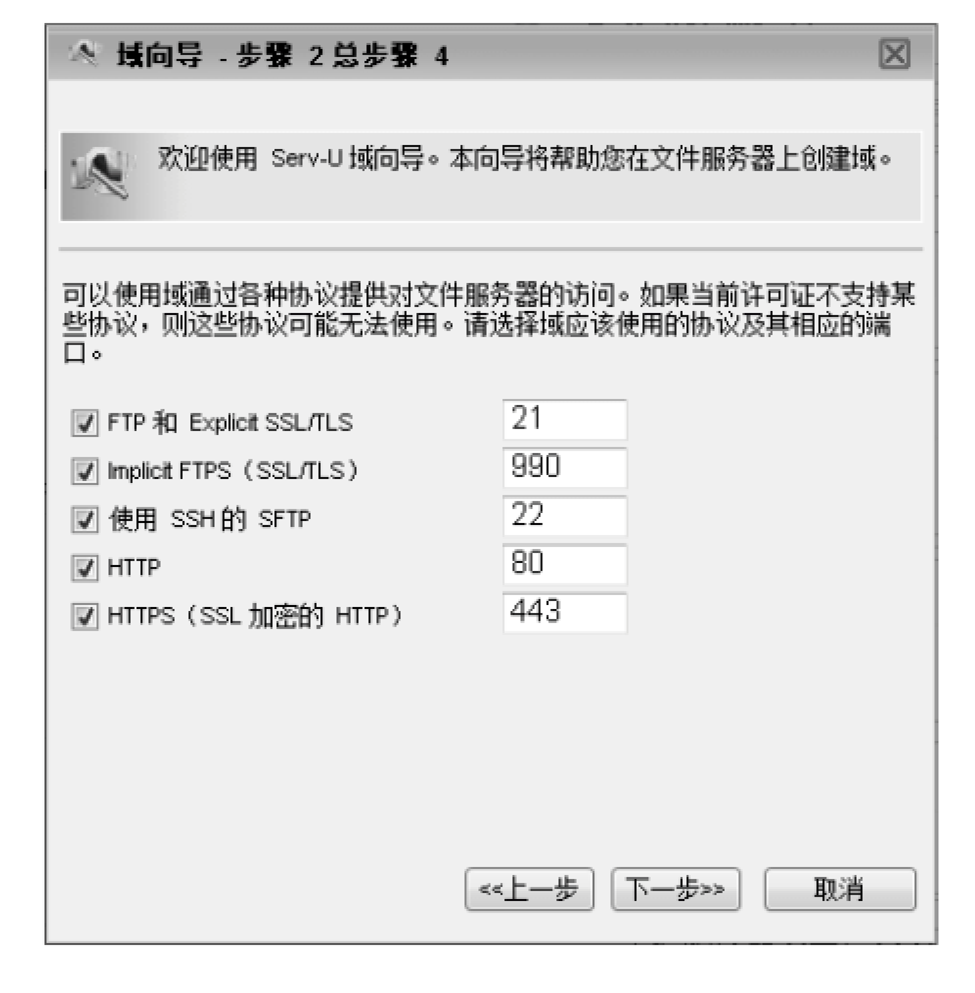


图 4.13　FTP 端口设置界面

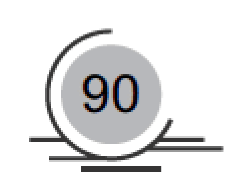


图 4.14　FTP 密码加密方式设置界面



如果用户只是在本机上测试，建议读者不要更改参数设置。

（4）填写域名称，然后单击“下一步”按钮直至设置完成。



（5）完成新建域操作后，返回管理控制台界面。单击“添加新用户”选项创建新用户。单击“添加”按钮进入用户添加界面，在该界面可以设置个人用户信息，即 FTP 服务端预设信息。选择“用户信息”选项卡，在该页面中填写登录 ID、全名、密码以及选择设置根目录、管理权限、SSH 公共密钥路径、账户类型、说明等信息。当用户填写完毕后，单击“保存”按钮即可完成用户添加操作。其中，添加用户的操作界面如图 4.15 所示。

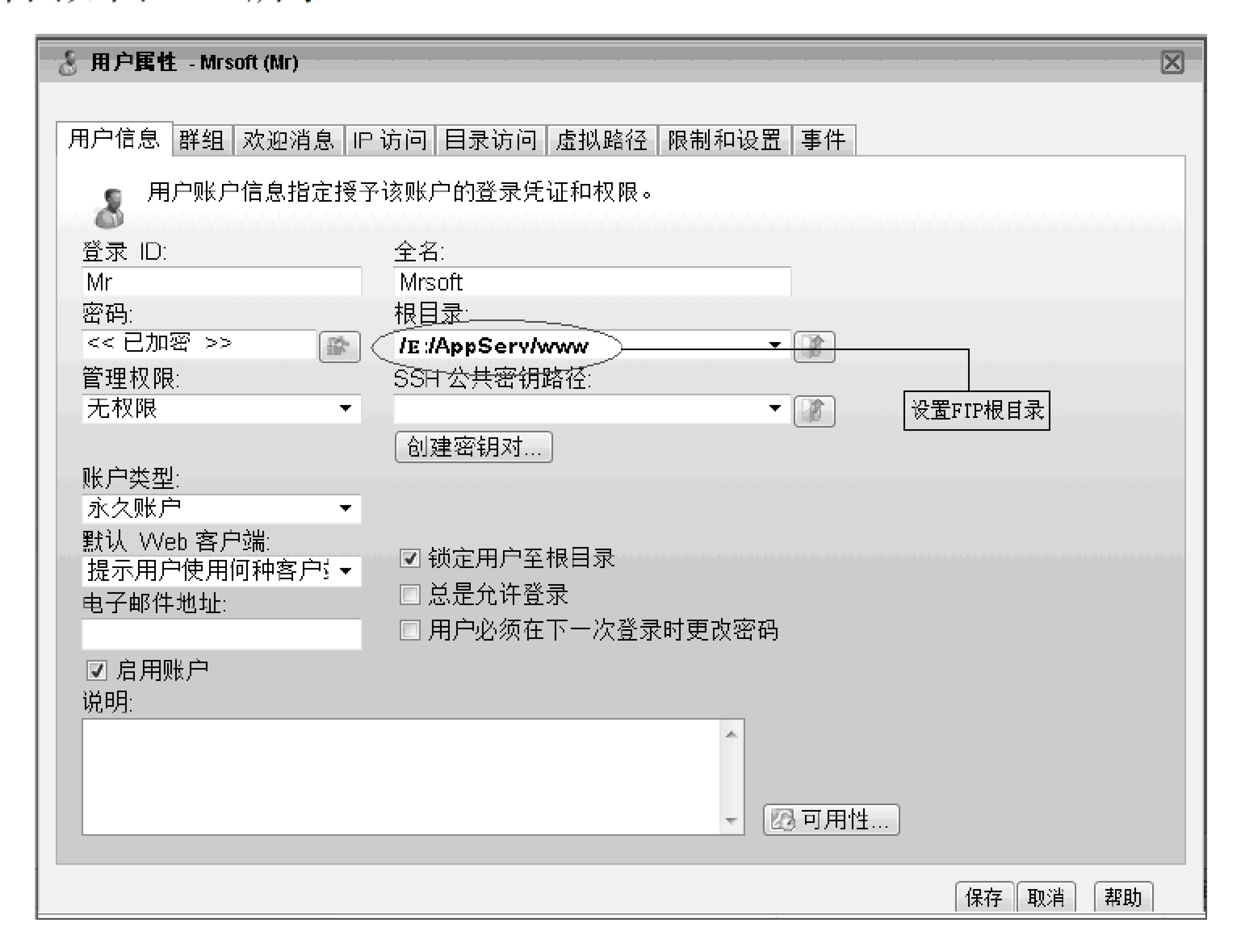


图 4.15　添加用户的操作界面

说明 建议读者将根目录设置成运行 PHP 环境的目录。这样在用户执行远程操作时，可以更方便地访问 PHP 服务器目录，从而使 FTP 管理系统发挥其更大的作用和优势。笔者在设置 Serv-U 服务端用户时，将 FTP 访问的根目录设置成 Appserv 集成环境所在的目录。

至此，FTP 服务器端的架设和参数设置介绍完毕。如读者需要对其他参数进行设置，建议参考官方提供的有关资料。由于篇幅所限，其他的设置参数不再一一列举。

4.3　用 户 登 录

4.3.1　用户登录功能概述

用户登录功能是 FTP 服务器端对用户输入的 IP 地址、用户名等 FTP 连接信息进行验证。与普通登录验证方式不同的是，用户在登录页面提交的验证信息需要与指定服务器的预设登录信息进行比对，如果用户输入的连接信息与预设信息匹配，则可以顺利连接 FTP 服务器，否则将不能连接 FTP 服务器。这样，就可以使用户安全地连接 FTP 服务器。登录页面的运行结果如图 4.16 所示。



图 4.16 登录页面的运行结果

用户登录功能实现包含 index.php（登录表单页）和 config.php（系统连接公共文件）2 个核心文件，它们用来完成对登录用户的验证操作。

4.3.2 登录页面设计

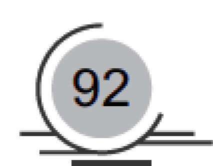
登录表单页（index.php）的结构相对简单，主要包含连接 FTP 服务器 IP 地址的文本框、用户名文本框、密码文本框和登录按钮。登录页面中使用的主要元素如表 4.1 所示。

表 4.1 登录页面中使用的主要元素

元 素 名 称	元素属性值	说 明
script	"js/check.php"	登录信息是否为空验证脚本
text	id="ip"name="ip"	连接服务器 IP 地址
text	id="username"name="username"	连接用户名
password	id="password"name="password"	连接用户密码
submit	name="login" value="登录"	“登录”按钮

在 FTP 管理系统的登录页中，通过 check.js 脚本中的 chklogin()函数对提交的登录信息进行验证。在 chklogin()函数中，通过获取 id 值为 login 的表单中的元素来判断文本框是否为空。如果表单元素中的文本框为空，则将焦点定位到该元素上。代码如下：

```
function chklogin(){
if(login.ip.value==""){
      alert("请填写连接 IP");login.ip.focus();return false;             //判断连接的 IP 文本框是否为空
}
if(login.username.value==""){
     alert("请填写连接用户名");login.username.focus();return false;   //判断连接的用户名是否为空
}
if(login.password.value==""){
      alert("请填写连接密码");login.password.focus();return false;     //判断连接的密码是否为空
}
login.submit();                                                          //提交 login 表单
}
```





4.3.3　系统连接公共文件

检测 FTP 登录用户信息的操作在 config.php 文件中完成。首先获取表单提交的登录数据，并且将其赋给指定的 Session 变量。代码如下：

```
<?php
if(isset($_POST['login'])){                         //判断是否以 POST 方式获取 login 传递参数
$_SESSION['ip'] = $_POST['ip'];                     //获取验证 IP 地址并赋给 Session 变量
$_SESSION['user'] = $_POST['username'];             //获取验证用户名并赋给 Session 变量
$_SESSION['pwd'] = $_POST['password'];              //获取验证密码并赋给 Session 变量
}
$ip = $_SESSION['ip'];                              //传递 Session 变量给相应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
```

然后，将获取的 FTP 地址、用户名称、密码作为连接参数，远程连接 IP 地址指向的服务器。如果连接成功，则显示 main.php 页面的其他内容；如果连接失败，则返回错误信息。相关代码如下：

```
$link = @ftp_connect($ip);                          //设置远程服务器连接标识
$login = @ftp_login($link, $user, $pwd);            //连接远程 FTP 服务器
if(!$login){                                        //判断用户是否登录成功
echo '错误，<a href="index.php">返回</a>';           //否则返回错误提示
exit();
}
?>
```

说明　本章系统将 config 文件目录下的 config.php 文件作为 FTP 连接的公共文件。为了避免重复书写 FTP 连接过程，在实现其他 FTP 功能时，只需通过 include 语句调用该公共文件完成与 FTP 服务器的连接即可。

4.4　根目录操作

4.4.1　根目录功能概述

用户在正常登录 FTP 管理系统后，可以遍历用户在 FTP 服务器中指向的系统根目录。用户可以单击根目录进入次级根目录，并可对根目录执行操作。进入 FTP 管理系统的根目录运行结果如图 4.17 所示。

其中，根目录的主要功能包括查看根目录名称、删除目录、创建文件目录、上传文件到根目录等。其主要是在 main.php（系统管理页）、mkdir.php（创建文件目录页）、rmkdir.php（删除文件处理页）等文件中实现。



图 4.17　FTP 管理系统的根目录运行结果

4.4.2　遍历文件根目录

遍历文件根目录，即用户进入 FTP 系统管理页面（main.php），应用 ftp_rawlist()函数结合 foreach 语句输出根目录中的所有文件目录名称。用户单击文件目录名称的超链接进入次级目录；如果用户想返回根目录或上级目录，单击“返回上级目录”超链接，即可返回父级目录。

1．遍历根目录名称

应用 ftp_pwd()、ftp_rawlist()函数获取路径名称、目录列表以及目录下的文件详细信息。代码如下：

```
<?php
$dirname = ftp_pwd($link);                        //返回当前目录名
if($dirname != '/'){                              //判断当前目录中是否存在“/”
$dirname .= '/';                                  //不存在则将路径名称赋值为“/”
}
$flist = ftp_rawlist($link, $dirname);            //返回当前目录下的详细列表
$sdir = array();                                  //更改变量$sdir 的数据类型
foreach($flist as $value){                        //循环输出目录详细列表
$array = explode(' ', rtrim($value));             //将去除右边空白字符的元素，以空格作为分割符分隔
$alen = count($array);                            //获取数组元素个数
$p = substr($array[0], 0, 1);                     //截取返回数组长度
if($p == 'd'){
if($array[$alen-1] != '.' && $array[$alen-1] != '..'){
$sdir[] = $array[$alen-1];
}
}
}
?>
```

将获取的当前目录名称、目录列表以及文件详细列表，在显示页中应用 foreach 语句循环输出。代码如下：

```
<div class="mainleft">
<table border="0" cellpadding="0" cellspacing="0">
<?php
//循环输出根目录的目录名称列表
foreach($sdir as $value)
echo'<tr>
<td height="10" width="30">  </td>
<td height="25" width="75" align="center" class="STYLE1">
<a href="main.php?dir='.$dirname.$value.'">'.substr($value,0,12).'</a>
</td>
<td width=50><a href="rmkdir.php?dir='.$dirname.$value.'">删除</a></td>
<td height="5" width="30">  </td>
</tr>';
?>
</table>
</div>
```

2．返回上级目录

返回上级目录是通过单击“返回上级目录”超链接来实现的。其实现原理是，根据 URL 地址中传递的目录路径 dir，应用 ftp_chdir()函数改变当前路径为指定路径。相关代码如下：

```
<?php
if(isset($_GET['dir'])){                                          //获得以 GET 方式传递的 URL 参数
ftp_chdir($link, $_GET['dir']);                                   //改变当前路径为指定路径
}
if(isset($_GET['up'])){                                           //判断用户是否单击“返回上级目录”超链接
ftp_cdup($link);                                                  //改变上级目录为指定路径
}
if(isset($_GET['down'])){                                         //判断用户是否单击文件目录名称
ftp_get($link, $_GET['down'], $_GET['down'], FTP_BINARY);         //进入下级文件目录
}
?>
```

4.4.3　删除文件目录

在系统的管理页面中，每一条显示文件目录名称后面都有一个“删除”超链接。单击“删除”超链接，将文件目录名称作为参数传递到 rmkdir.php 文件，完成文件目录的删除操作。相关代码如下：

```
<?php
session_start();                                                  //开启全局会话
$ip = $_SESSION['ip'];                                            //将 Session 变量中的数据赋值给对应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
$link = @ftp_connect($ip);                                        //连接到指定 FTP 地址
```

```
$login = @ftp_login($link, $user, $pwd);                          //连接服务器
$file = $_GET['dir'].$_GET['file'];                               //设置删除文件位置
if(ftp_delete($link, $file)){                                     //判断文件是否被删除
    echo '<script>alert("删除文件成功！");history.back();</script>';  //删除成功提示
}else{
    echo '<script>alert("删除文件失败！");history.back();</script>';  //删除失败提示
}
ftp_close($link);                                                 //关闭 FTP 连接
?>
```

4.4.4　在根目录中创建新目录

单击系统管理页面中的“创建目录”超链接，跳转到 mkdir.php 页面，填写要在根目录中创建的新目录名称，最终将数据提交到 mkdir_chk.php 文件中，完成新目录的创建。

在 mkdir.php 文件中，完成目录名称的添加，其代码如下：

```
<link href="css/mkdir.css"  type="text/css" rel="stylesheet" />
<script src="js/check.js"></script>
<form id="mkdir" action="mkdir_chk.php?dir=" method="post" onSubmit="return chkname();">
<input id="dirname" class="inpt1" type="text" name="filename" />
<input  type="submit" value="提交" />
<input  type="reset" value="重置" />
</form>
```

在 mkdir_chk.php 文件中，获取表单提交的目录名称，应用 ftp_mkdir()函数在 FTP 服务器根目录中创建新文件目录。代码如下：

```
<?php
session_start();
$ip = $_SESSION['ip'];                              //将 Session 变量中的数据赋值给对应变量
$user = $_SESSION['user'];
$pwd = $_SESSION['pwd'];
if(!empty($_GET["dir"])){                           //判断是否通过 GET 方式获取文件目录路径
$path=$_GET["dir"];                                 //将路径名称赋值给对应变量
}else{
$path="";                                           //否则为根目录
}
$dirname=$path.$_POST["filename"];                  //指定创建文件路径
$link = @ftp_connect($ip);                          //连接指定 FTP 地址
$login = @ftp_login($link, $user, $pwd);            //连接 FTP 服务器
$result = ftp_mkdir($link, $dirname);               //创建文件目录
if($result) {                                       //判断文件目录是否被创建
  echo '<script>alert("目录'.$dirname.'\n 创建成功！");window.close();;</script>';    //提示创建成功
}else {
  echo '<script>alert("目录'.$dirname.'\n 创建成功！");history.back();</script>';     //提示创建失败
}
  ftp_close($link);                                 //关闭 FTP 连接
?>
```



4.4.5　上传文件到根目录下

在系统管理页中，单击“上传文件”超链接，跳转到 upfile.php 页面，选择要上传的本地文件，将数据提交到 upfile_chk.php 文件中，实现将本地文件上传到服务器指定的文件夹下。

在 upfile.php 文件中创建表单并添加文件域，将本地文件提交到 upfilechk.php 文件中。代码如下：

```
<form action="upfile_chk.php" method="post" enctype="multipart/form-data">
<input class="inpt1"   type="file" name="file" />
<input class="inpt1" type="submit" value="提交" />
</form>
```

在 upfile_chk.php 文件中获取表单中提交的本地文件信息，应用 ftp_put()函数将文件上传到服务器的根目录中。相关代码如下：

```
<?php
session_start();                                                    //开启全局会话
include_once("config/config.php");                                  //加载公共连接文件
$up=$_FILES["file"]["tmp_name"];                                    //获取上传文件
$file=$_FILES["file"]["name"];                                      //定义上传文件名称
if(ftp_put($link,$file,$up,FTP_BINARY)){                            //上传文件到服务器根目录中
echo '<script>alert("上传'.$file.'\n 成功！ ");window.close();</script>';  //提示上传成功
}else {
echo '<script>alert("上传'.$file.'\n 失败！ ");history.back();</script>';   //提示上传失败
}
ftp_close($link);                                                   //关闭 FTP 连接
?>
```

4.5　子目录操作

4.5.1　子目录功能概述

用户单击根目录文件会进入次级目录，并在主窗体显示该文件目录中的子目录和目录下的所有文件。子目录中部分操作和根目录操作大同小异。但是用户进入子目录后，不但可以操作子文件目录，还可以对文件目录下的文件执行多种操作。进入子目录的运行结果如图 4.18 所示。

用户对子目录执行的操作包括查看子目录文件，删除子目录，在子目录中创建新文件目录，上传文件到当前子目录，更改子目录下的文件名称，移动文件到其他目录中，以及下载远程文件到本地计算机中。对子目录的操作，主要是在 main.php（系统管理页）、removefile.php（删除文件处理页）、movefile.php（移动文件处理页）等文件中完成的。

说明 由于目录的创建和上传文件等功能与根目录操作大体相同，所以在后面章节中不再进行讲解。下面只对子目录中特有的一些功能进行讲解。



图 4.18　子目录的运行结果

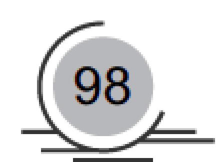
4.5.2　浏览子目录下的文件

用户登录 FTP 管理系统后，单击任意根目录进入次级目录，在主窗体同时会显示子目录中包含的所有文件的相关信息。其中，显示内容包括文件名称、文件大小、最后修改时间等。另外，用户也可以对该目录下的文件执行移动、更改名称等操作。在 main.php 文件中，遍历子目录中所有文件相关信息的关键代码如下：

```
<?php
$dirname = ftp_pwd($link);                        //返回当前目录名
if($dirname != '/'){                              //判断当前目录中是否存在“/”
$dirname .= '/';                                  //不存在则将路径名称赋值为“/”
}
$sfile = ftp_nlist($link, $dirname);              //返回当前根目录列表
if(!$sfile){                                      //判断根目录下是否为空
   $sfile = array();                              //更改变量$sfile 的数据类型
}
$flist = ftp_rawlist($link, $dirname);            //返回当前目录下的详细列表
?>
```

通过 foreach 语句循环输出子目录中的详细内容。关键代码如下：

```
<table border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#BAE8BC">
<tr>
   <td height="35"   colspan="6" bgcolor="#FFFFFF"><div style=" margin:auto 20px;text-align:right;" >
    <input type="button" value="" class="btn3" onclick="javascript:Wopen=open('path_mkdir.php?
dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no');">
<input type="button" class="btn4" onclick="javascript:Wopen=open('path_upload.php?
```



```
dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no');"></div>
</td>
</tr>
<tr>
  <td height="28" align="center" bgcolor="#FFFFFF">文件名称</td>
<td align="center" bgcolor="#FFFFFF">文件大小</td>
<td align="center" bgcolor="#FFFFFF">最后修改时间</td>
  <td align="center" bgcolor="#FFFFFF">更改名称</td>
<td align="center" bgcolor="#FFFFFF">删除文件</td>
  <td align="center" bgcolor="#FFFFFF">移动文件</td>
</tr>
<?php
foreach($sfile as $value){                              //循环输出文件列表
?>
<tr>
<td width="250" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('download.php?dir=<?php echo $dirname; ?>
&filename=<?php echo $value;?>','','height=220,width=350,scrollbars=no');"><?php echo $value; ?></a>
</td>
<td width="70" height="30" align="center" bgcolor="#FFFFFF"><?php echo round(ftp_size($link,$value)/1024,
2);?> K
</td>
  <td width="200" height="30" align="center" bgcolor="#FFFFFF">
<?php echo date('Y-m-d H:i:s',ftp_mdtm($link,$value)); ?>
</td>
<td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('edit.php?file=<?php echo $value; ?>
&dir=<?php echo $dirname; ?>','','height=220,width=350,scrollbars=no'); ">更改</a>
</td>
     <td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="removefile.php?file=<?php echo $value; ?>&dir=<?php echo $dirname;?>">删除</a>
</td>
<td width="60" height="30" align="center" bgcolor="#FFFFFF">
<a href="#" onclick="javascript:Wopen=open('movefile.php?dir=<?php echo $dirname; ?>
&filename=<?php echo $value;?>','','height=220,width=350,scrollbars=no');">移动</a></td>
</tr>
<?php
}                                                       //foreach 语句结束
?>
</table>
```

4.5.3　远程文件下载

作为一款 FTP 管理工具，下载远程服务器中的文件这一功能必不可少，与普通 Web 程序下载功能不同，FTP 服务器上的文件并不能直接通过单击超链接形式下载。因为 FTP 服务器指定的目录不一定是 Web 服务器或提供服务的环境目录，所以这里需要通过 ftp_get()函数来执行远程文件下载操作。

单击“文件名称”超链接，进入 download.php 文件下载页面，填写正确的本地下载路径，单击“下

载”按钮，将数据提交到 download.php 文件中，将远程服务器中的文件下载到本地。download.php 文件的代码如下：

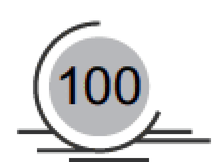
```
<?php
session_start();                                        //开启全局会话
include_once("config/config.php");                      //加载公共连接文件
if(!empty($_POST["savepath"])){                         //判断是否存在下载路径
$absopath=$_POST["savepath"];                           //获取下载路径
$dir=$_GET["dir"];                                      //获取下载的文件路径
$dwname=$_GET["filename"];                              //获取下载的文件名称
$file=$dir.$dwname;                                     //定义下载文件
$down=$absopath.$dwname;                                //定义下载路径
if(ftp_get($link,$down,$file,FTP_BINARY)) {             //判断是否从 FTP 服务器中下载文件
   echo '<script>alert("文件'.$file.'\n 下载成功！ ");window.close();</script>';
}else {
      echo '<script>alert("文件'.$file.'\n 下载！ ");history.back();</script>';
}
ftp_close($link);                                       //关闭 FTP 连接
}else{                                                  //如果下载路径不存在，则向用户展示下载提示框
?>
<script src="js/check.js"></script>
<!--省略部分代码-->
<form id="download" action="#" method="post" onsubmit="return chkdown();">
<input id="down" class="ipt1" type="text" name="savepath" />
<p><span style=" color:#FF0000;">*请填写有效路径（如 C:\Leon\Appserv\)</span></p>
<input class="btn" type="submit" value="下载" />  <input class="btn" type="reset" value="重置" />
</form>
<?php
}                                                       //结束选择判断
?>
```

说明 （1）远程文件下载操作中，路径输入的 form 表单与远程文件的下载操作均存储于 down.php 文件中。

（2）远程文件下载功能中输入的下载路径要按照路径规范书写。如图 4.19 所示定义的就是一个规范的文件下载路径。



图 4.19　定义规范的远程文件下载路径





4.5.4　文件移动

用户可以选定文件并将其作为对象移动到 FTP 的其他目录中。单击文件详细列表中的“移动”超链接，进入移动路径的填写页面 movefile.php 中，添加完整的移动路径，单击“移动”按钮，将数据提交到本页，完成文件的移动操作。movefile.php 文件的代码如下：

```
<?php
session_start();                                            //开启全局会话
include_once("config/config.php");                          //加载公共连接文件
$filename=$_GET["dir"].$_GET["filename"];                   //定义文件路径
$fname=$_GET["filename"];                                   //获取文件名称
if(empty($_POST["movepath"])){                              //判断路径是否存在
?>
<form id="movefile" action="#" method="post" onSubmit="return chkpath();">
将<?php echo $filename; ?>移动到：
<p>
<center>
  <input class="inpt1" id="path" type="text" name="movepath">
</center>
<p>
<center>
<span class="notice">*请填写正确的路径。"如/Appserv/"</span>
</center>
<p>
<center>
<input class="btn2" type="submit" value="移动文件">
</center>
</form>
<?php }else{
$newpath=$_POST["movepath"].$fname;                         //如果存在，则执行移动操作
if(ftp_put($link,$newpath,$filename,FTP_BINARY)){           //将文件移动到指定文件目录
unlink($filename);                                          //删除原来位置的文件
   echo '<script>alert("移动'.$file.'\n 到'.$_POST[movepath].'成功！");window.close();</script>';
}else {                                                     //提示移动成功
  echo '<script>alert("移动'.$file.'\n 到'.$_POST[movepath].'失败！");history.back();</script>';
}                                                           //提示移动失败
ftp_close($link);                                           //关闭 FTP 连接
}?>
```

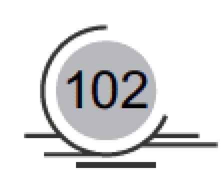
4.5.5　更改文件名称

在 FTP 管理系统中，用户不但可以执行访问、查看、下载文件等操作。同时如果在移动文件过程中，文件名称冲突可能会导致移动失败，所以开发了文件名称更改的功能。单击文件详细信息列表中指定文件后的“更改”超链接，跳转到 edit.php 页面中，添加新的文件名称，单击“更改”按钮，将数据提交到本页，完成对文件名称的更改操作。edit.php 文件的关键代码如下：

```
<!--创建添加文件名称的表单-->
<table>
<!--省略部分代码-->
<td width="770" align="center"><table   border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td align="center" class="STYLE1">
<form id="form1" name="form1" method="post"
action="edit.php?dir=<?php echo $_GET['dir'].'&file='.$_GET['file'] ?>" onSubmit="return chkamend();">
                <p>原文件名为：<?php echo $_GET['file']; ?></p>
                <p>输入文件名：
<input id="news" type="text" name="new" />
                  
                <input type="submit" name="Submit" value="更改" />
               </p>
                <p>注意：要输入文件的全名，包括扩展名</p>
              </form>
            </td>
```

获取 URL 参数，应用 ftp_rename()函数更改文件名称。代码如下：

```
<?php
//获取表单提交的文件名称，完成对指定文件名称的更改操作
session_start();                                          //开启全局会话
include_once("config/config.php");                        //加载公共连接文件
$file = $_GET['dir'].$_GET['file'];                       //获取文件完整路径
if($_POST){                                               //判断是否存在以 POST 方式提交的数组
$new = $_GET['dir'].$_POST['new'];                        //获取新命名文件的完整路径
$result = ftp_rename($link, $file, $new);                 //更改名称
if($result)                                               //判断是否更改成功
{
echo '<script>alert("文件'.$_GET['file'].'\n 改名为\n'.$_POST['new'].'\n 成功");window.close();</script>';
}                                                         //提示更改成功
else{
echo '<script>alert("文件'.$_GET['file'].'\n 改名为\n'.$_POST['new'].'\n 失败");history.back()";</script>';
}                                                         //提示更改失败
}
?>
```



说明 更改文件名称时必须添加扩展名，如 index.php 而不是 index。如果更改时未添加扩展名，可能导致程序无法正常运行。

4.6 技 术 提 炼

4.6.1 FTP 文件操作技术

对 FTP 文件的操作，应用的是 PHP 内置的 FTP 文件传输函数库中的函数。下面对其中常用的函



数进行详细讲解。

（1）ftp_connect()函数

该函数用来建立一个新的 FTP 连接，如果成功，则返回一个连接标识，否则返回 false。该函数的语法如下：

```
resource ftp_connect( string host [, int port [, int timeout]] )
```

☑ host：必要参数。FTP 服务器的地址，后面不应以斜线结尾，前面也不需要以 ftp://开头。
☑ port：可选参数。FTP 服务器的端口，默认端口为 21。
☑ timeout：可选参数。设置网络传输的超时时间限制，默认值为 90 秒，适用于 PHP 4.2.0 及以上版本。

下面示例应用 ftp_connect()函数改变 FTP 服务器上一个文件的属性。代码如下：

```
<?php
$ftp_server = "127.0.0.1";
$link = ftp_connect($ftp_server) or die("不能连接 FTP 服务器");            //建立 FTP 连接
if($link){
    echo "连接 FTP 服务器成功";
}
ftp_close($link);                                                          //关闭 FTP 连接
?>
```

（2）ftp_login()函数

利用该函数可以登录到 FTP 服务器，如果成功返回 true，否则返回 false。该函数的语法如下：

```
bool ftp_login( resource ftp_stream, string username, string password )
```

☑ ftp_strea：必要参数。FTP 连接的标识符。
☑ username：必要参数。FTP 服务器的用户名。
☑ password：必要参数。FTP 服务器的密码。

下面示例应用 ftp_login()函数登录到 FTP 服务器。代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");                           //登录 FTP 服务器
if($login_result) {
    echo "登录成功";
}else {
    echo "登录失败";
}
ftp_close($link);                                                          //关闭 FTP 连接
?>
```

（3）ftp_chdir()函数

利用该函数可以在 FTP 服务器上改变当前的目录为指定目录，成功返回 true，否则返回 false。该

函数的语法如下：

```
bool ftp_chdir( resource ftp_stream, string directory )
```

应用该函数切换当前目录为指定目录的代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");                        //连接指定地址
$login_result = ftp_login($link, "mr", "mrsoft");        //连接 FTP 服务器
ftp_chdir($link, "myfile");                              //改变目录
echo "当前目录为".ftp_pwd($link);                        //输出当前的目录名
ftp_close($link);                                        //关闭 FTP 连接
?>
```

（4）ftp_mdtm()函数

该函数用来检查指定文件的最后修改时间，并以 UNIX 时间戳的方式返回，如果发生错误，或文件不存在，则返回-1。某些 FTP 服务器可能会不支持这个特性。该函数的语法如下：

```
int ftp_mdtm( resource ftp_stream, string remote_file )
```

☑ ftp_stream：必要参数。指定 FTP 连接的标识符。

☑ remote_file：必要参数。设置 FTP 服务器的名称。

应用该函数检查文件修改时间的代码如下：

```
<?php
$file = "myfile/mingri.php";                             //设置文件名称
$link = ftp_connect("127.0.0.1");                        //连接指定地址
$login_result = ftp_login($link, "mr", "mrsoft");        //连接 FTP 服务器
$result = ftp_mdtm($link, $file);                        //检查指定文件的最后修改时间
if($result == -1) {                                      //判断查询的文件是否存在
    echo "查询失败";
}else {
    echo $file."的最后修改时间：".date('Y-m-d H:i:s', $result);    //输出最后修改时间
}
ftp_close($link);                                        //关闭 FTP 连接
?>
```

（5）ftp_nlist()函数

该函数用来返回给定目录的文件列表，如果成功，则返回给定目录下的文件名组成的数组，否则返回 false。该函数的语法如下：

```
array ftp_nlist ( resource ftp_stream, string directory )
```

应用 ftp_nlist()函数返回给定目录的文件列表的代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");                        //连接指定地址
```

```
$login_result = ftp_login($link, "mr", "mrsoft");                        //连接 FTP 服务器
$result = ftp_nlist($link, "myfile");                                    //返回给定目录的文件列表
print_r($result);                                                        //输出数组结果集
ftp_close($link);                                                        //关闭 FTP 连接
?>
```

（6）ftp_rename()函数

该函数将 FTP 服务器上的一个文件或目录改名，如果成功则返回 true，否则返回 false。该函数的语法如下：

```
bool ftp_rename( resource ftp_stream, string oldname, string newname )
```

☑ ftp_stream：必要参数。FTP 连接的标识符。

☑ oldname：必要参数。旧的名称。

☑ newname：必要参数。新的名称。

应用该函数实现文件或目录改名的代码如下：

```
<?php
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");
$rename1 = ftp_rename($link, "myfile", "mynewfile");                     //将 myfile 文件目录重新命名
if($rename1){                                                            //判断目录名称是否被更改
     echo "更改目录名成功";                                              //判断文件目录名称是否被更改
}else{
     echo "更改目录名失败";
}
$rename2 = ftp_rename($link, "mr.php", "mingri.php");                    //将 mr.php 文件重新命名
if($rename2){                                                            //判断文件名称是否被更改
     echo "更改文件名成功";
}else{
     echo "更改文件名失败";
}
ftp_close($link);                                                        //关闭 FTP 连接
?>
```

（7）ftp_get()函数

该函数用来下载 FTP 服务器上的文件，如果成功返回 true，否则返回 false。该函数的语法如下：

```
bool ftp_get( resource ftp_stream, string local_file, string remote_file, int mode [, int resumepos] )
```

ftp_get()函数的参数说明如表 4.2 所示。

表 4.2　ftp_get()函数的参数说明

参　数	说　明
ftp_stream	必要参数。FTP 连接的标识符
local_file	必要参数。保存到本地的文件名

续表

参　数	说　明
remote_file	必要参数。FTP 服务器上的文件名称
mode	必要参数。传送模式参数，为 FTP_ASCII（文本模式）或 FTP_BINARY（二进制模式）中的一个
resumepos	可选参数。仅适用于 PHP 4.3.0 以上版本

应用该函数下载指定文件的代码如下：

```
<?php
$down = "mr.php";                                    //设置保存到本地文件名
$file = "myfile/mingri.php";
$link = ftp_connect("127.0.0.1");                    //建立 FTP 连接
$login_result = ftp_login($link, "mr", "mrsoft");
if(ftp_get($link,$down,$file,FTP_BINARY)) {          //判断是否从 FTP 服务器中下载文件
    echo "下载 $file 成功";
}else {
    echo "下载 $file 失败";
}
ftp_close($link);                                    //关闭 FTP 连接
?>
```

（8）ftp_put()函数

该函数用来上传一个已经打开的文件到 FTP 服务器，如果上传成功，返回 true，否则返回 false。该函数的语法如下：

```
bool ftp_put( resource ftp_stream, string remote_file, resource handle, int mode [, int startpos] )
```

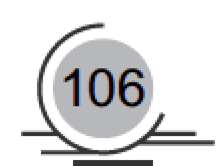
ftp_put()函数的参数说明如表 4.3 所示。

表 4.3　ftp_put()函数的参数说明

参　数	说　明
ftp_stream	必要参数。FTP 连接的标识符
remote_file	必要参数。本地已经打开文件的句柄
handle	必要参数。FTP 服务器上的文件名称
mode	必要参数。传送模式参数，为 FTP_ASCII（文本模式）或 FTP_BINARY（二进制模式）中的一个
startpos	可选参数。仅适用于 PHP 4.3.0 以上版本

应用该函数上传指定文件的代码如下：

```
<?php
$up = "soft.php";                                //本地上传文件名
$file = "myfile/mrsoft.php";                     //指定服务器中的目录名称
$link = ftp_connect("127.0.0.1");
$login_result = ftp_login($link, "mr", "mrsoft");
if(ftp_put($link, $file, $up,FTP_BINARY)) {      //判断是否上传文件到指定目录中
    echo "上传 $file 成功";
```



```
} else {
    echo "上传 $file 失败";
}
ftp_close($link);                              //关闭 FTP 连接
?>
```

此外，本 FTP 管理系统中还存在其他的 FTP 函数，由于这些函数并非本章重点，所以不再一一介绍。

4.6.2　循环输出技术

所有程序设计中都包含循环控制语句，用于反复多次处理问题。在 FTP 管理系统中，遍历指定文件目录下的内容，应用的是 ftp_nlist()函数，其返回值为数组，所以选择 foreach 语句循环输出文件目录下的内容。foreach 语句的特点就是操作数组，其语法如下：

```
foreach (array_expression as $value)
    statement
或者
foreach (array_expression as $key => $value)
    statement
```

其中，第一种格式遍历给定的 array_expression 数组。每次循环中，当前单元的值被赋给$value，并且数组内部的指针向前移一步（因此下一次循环中将会得到下一个单元）。

第二种格式的实现功能与第一种格式相同，只是在输出上略有不同，即元素循环输出时，$key 被作为指定元素的键名一同输出。


注意　foreach 语句不支持“@”来禁止错误信息。

例如，在本系统中应用 foreach 语句来循环输出子目录中文件相关信息的代码如下：

```
<?php
$dirname = ftp_pwd($link);
if($dirname != '/'){
$dirname .= '/'; $flist = ftp_rawlist($link, $dirname);
$sdir = array();
foreach($flist as $value){
$array = explode(' ', rtrim($value));
$alen = count($array);
$p = substr($array[0], 0, 1);
if($p == 'd'){
if($array[$alen-1] != '.' && $array[$alen-1] != '..'){
$sdir[] = $array[$alen-1];
}
}
}
} foreach($sdir as $value)
```

```
echo'<tr>
   <td height="10" width="30">  </td>
<td height="25" width="75" align="center" class="STYLE1">
<a href="main.php?dir='.$dirname.$value.'">'.substr($value,0,12).'</a> </td>
   <td width=50><a href="rmkdir.php?dir='.$dirname.$value.'">删除</a></td>
   <td height="5" width="30">  </td>
</tr>';
?>
```

4.6.3 应用 CSS 设置屏幕滚动条技术

FTP 管理系统的结构和一般连接数据库的 Web 程序不同，其在页面返回的结果依据实际目录中的文件，而一般连接数据库的 Web 程序则是通过连接数据库并应用数据库查询语句返回查询结果。连接数据库的 Web 程序可以通过分页来实现数据的分段显示，而 FTP 管理系统则很难实现这样的分段显示，面对文件目录中数目很多的目录，返回的文件结果将会撑开整个屏幕。所以，这里应用 CSS 相关技术设置屏幕的滚动条，实现文件目录下数据的滚动输出，即当数目大于所给边框长度时，系统的返回结果将会以滚动条的形式输出结果。

自动判断并设置滚动条的 CSS 的关键语法如下：

```
overflow:auto;                          //设置自动判断并设置滚动条
```

另外，也可以设置横向滚动条和纵向滚动条。语法格式如下：

```
overflow-x:auto;                        //设置横向自动判断并设置滚动条
overflow-y:hidden;                      //设置纵向自动判断并隐藏滚动条
```

自动设置滚动条的一般过程如下：

（1）设置 div 布局层，并为该层设置高度。相关代码如下：

```
<div class="content">
<!--HTML 代码部分-->
</div>
```

（2）在<head></head>标签中，添加<style></style>属性标签，并设置 class 为 content 的 CSS 样式表为自动调节高度和宽度。相关代码如下：

```
<head>
<style   type="text/css">
content{overflow:auto;height:200px;}
</style>
<head>
```

这样，当类为 content 的 div 容器中内容高度超过 200px 值时，在 div 的边框会自动显示滚动条。用户可以通过拖动滚动条来查看类为 content 的 div 中所显示的全部内容。其运行效果如图 4.20 所示。

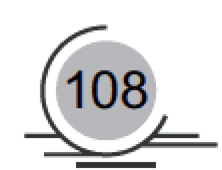




图 4.20　应用 CSS 样式设置滚动条

4.7　本 章 小 结

本章通过一个完整的 FTP 管理系统使读者详细了解使用 FTP 进行文件上传与下载的操作方法。通过 Serv-U 软件设置一台 FTP 服务器，用户或其他使用者就能够通过互联网或者同一网络上的任何一台计算机与 FTP 服务器连接，执行文件或目录的复制、移动、创建和删除等操作。

第 5 章

假日公告管理模块

（PHP+jQuery+MySQL 实现）

上网浏览网页时，经常会遇到打开网页后弹出一个小窗口的情况，如今很多大企业门户网站，都通过这种弹出式窗口发布一些公告或者假日问候等。本章将开发一个假日公告管理模块。

通过阅读本章，可以学习到：

- 数据库的连接与数据操作
- 通过超链接传递值
- 单文件上传
- 多图片上传
- 读取指定目录中的目录及文件
- 使用 jQuery 控制弹出公告信息



5.1　假日公告管理模块概述

假日公告管理模块实现发布及管理假日公告信息的功能，当要发布假日公告时，在假日公告发布模块中上传公告文件及图片即可，让用户及时了解公司的工作时间，并且可以第一时间为用户送上假日祝福。假期过后，可以通过假日公告取消模块取消发布的假日公告，然后通过假日公告管理模块将公告信息删除。

5.1.1　功能结构

假日公告管理模块的功能结构如图 5.1 所示。

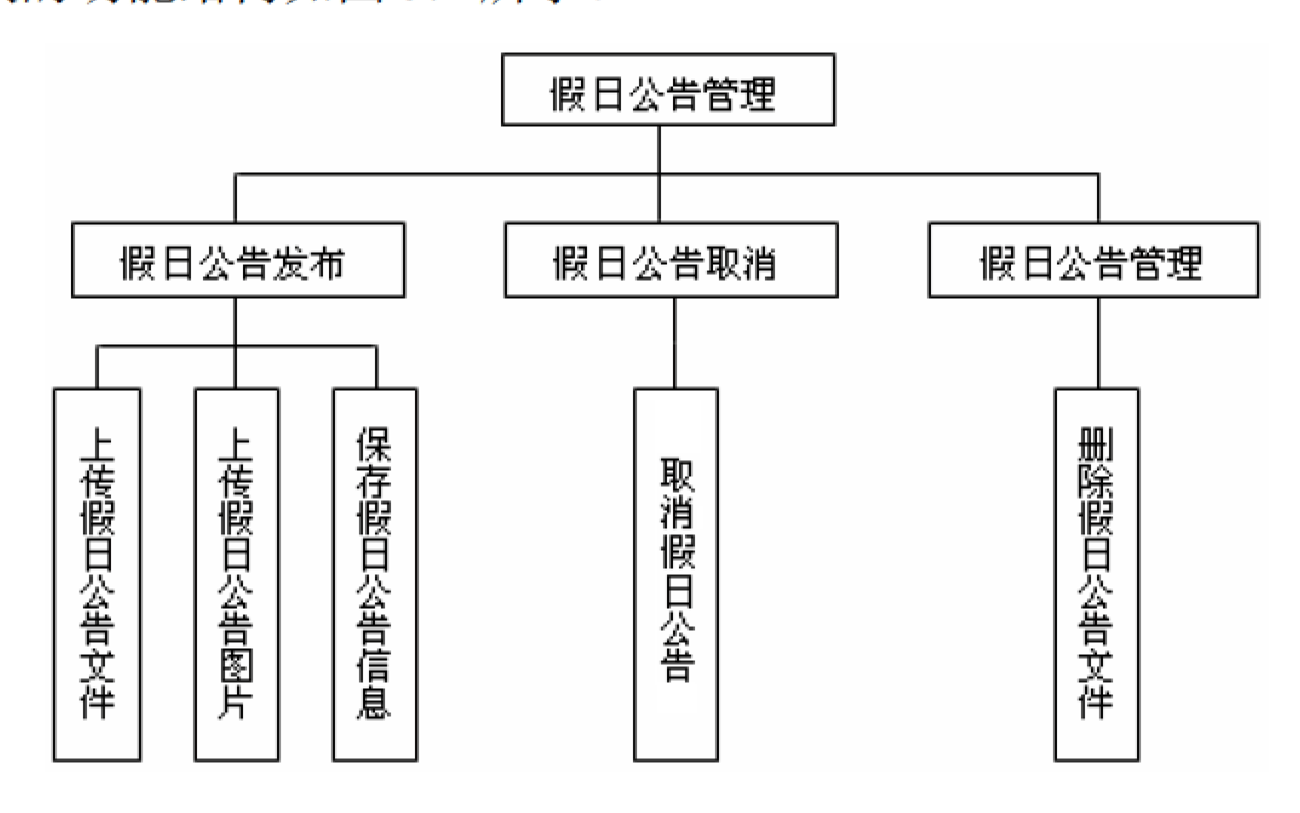


图 5.1　假日公告管理模块的功能结构

5.1.2　程序预览

为了让读者更好地了解假日公告管理模块的功能，下面列出假日公告管理模块中一些功能的运行效果，供广大读者参考。

运行网站，弹出公告信息页面，效果如图 5.2 所示。



图 5.2　公告信息页面效果

假日公告管理模块的主页面运行效果如图 5.3 所示。

登录页面运行效果如图 5.4 所示。



图 5.3　假日公告管理模块的主页面



图 5.4　登录页面运行效果

假日信息发布页面运行效果如图 5.5 所示。



图 5.5　假日信息发布页面运行效果

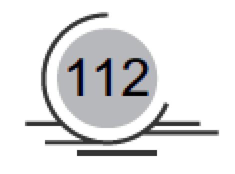
假日公告信息取消页面运行效果如图 5.6 所示。假日公告信息管理页面运行效果如图 5.7 所示。



图 5.6　假日公告信息取消页面运行效果



图 5.7　假日公告信息管理页面运行效果



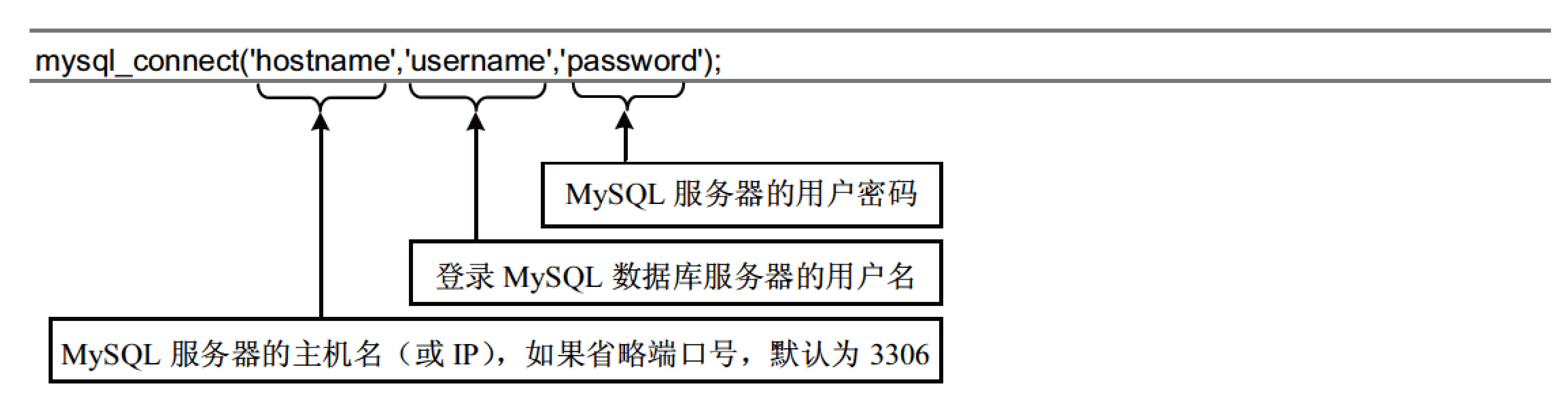


5.2　关 键 技 术

5.2.1　数据库的连接与数据操作

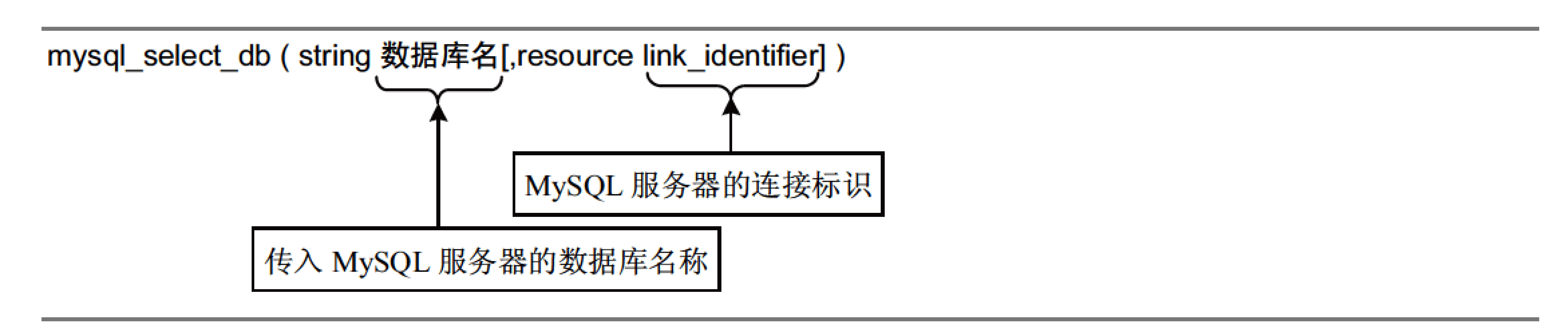
1．数据库的连接

（1）要操作 MySQL 数据库，首先必须与 MySQL 服务器建立连接。连接 MySQL 服务器应用的是 mysql_connect()函数，其语法如下：



该函数的返回值用于表示这个数据库连接。如果连接成功，则函数返回一个资源，为以后执行 SQL 指令做准备。

（2）在连接到 MySQL 数据库服务器之后，接下来使用 mysql_select_db()函数选择数据库。该函数的语法如下：



mysql_query()函数的语法如下：

```
mysql_query("use 数据库名"[,resource link_identifier]);
```

如果没有指定连接标识符，则使用上一个打开的连接。如果没有打开的连接，本函数将无参数调用 mysql_connect()函数来尝试打开一个数据库并使用。其后的 mysql_query()函数调用都会作用于活动数据库。

例如，在本模块中，与 MySQL 数据库服务器中的 db_jiari 数据库建立连接。conn\conn.php 文件的代码如下：

```
$conn=mysql_connect("127.0.0.1","root","111");              //连接数据库服务器
mysql_select_db("db_jiari",$conn);                          //连接数据库
mysql_query("set names utf8");                              //设置编码
```

2. MySQL 数据操作

对数据库中数据的操作主要是使用 mysql_query()函数，本模块主要使用 mysql_query()函数实现添加和删除数据的操作。

（1）添加数据

当用户发布假日公告信息时，将向数据库中添加一条假日信息。promulgation_ok.php 文件的关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_POST['name']) and $_POST['Submit']=="发布"){
	$insert=mysql_query("insert into tb_01(name,perform) values('".$_POST['name']."','".$_POST['radio']."')",
$conn);                                         //添加数据
…                                               //省略部分代码
}else{
	echo "<script>alert('请选择上传文件！');window.location.href='promulgation.php';</script>";
}
?>
```

（2）删除数据

当用户取消假日公告时，将删除数据库中的假日信息。update.php 文件的代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_GET['abrogation'])){
	$update=mysql_query("delete from tb_01 where id='".$_GET['abrogation']."'",$conn);
	if($update){
		echo "<script>alert('假日信息取消成功！');window.location.href='cancel.php';</script>";
	}else{
		echo "<script>alert('假日信息取消失败！');window.location.href='cancel.php';</script>";
	}
}
?>
```

5.2.2 通过超链接传递值

通过超链接传递值主要使用<a>…标记，其语法格式如下：

```
<a href=URL name=name target=target>链接文字</a>
```

<a>…标记的属性如下。

☑ href 属性：指定所链接文件的 URL 路径。该路径可以是相对路径，也可以是绝对路径。

☑ name 属性：指定页面的锚点名称，如果想要链接到对应的锚点位置，需要在锚点名称前添加一个“#”字符。





☑ target 属性：指定要打开的链接所使用的浏览器窗口名称，可以使用自定义的窗口名称，也可以使用下面 4 个内置的窗口名称，其中前两个比较常用。
  ➢ _self：在当前窗口中打开链接文件，是默认值。
  ➢ _blank：开启一个新的窗口打开链接文件。
  ➢ _parent：在父级窗口中打开文件，常用于框架页面。
  ➢ _top：在顶层窗口中打开文件，常用于框架页面。

本模块主要在取消公告及删除公告文件中使用了超链接。

假日公告信息取消页中的“取消”超链接的代码如下：

```
<a href="update.php?abrogation=<?php echo $array['id'];?>">取消</a>
```

假日公告信息管理页中的“删除”超链接的代码如下：

```
<a href='delete.php?catalog=".urlencode($catalog)."&filename=" . urlencode(getcwd ()) . "' title='删除目录或者文件' >删除</a>
```


说明　获取超链接参数传递的值，应用的是$_GET[]全局变量。

5.2.3　单文件上传

本模块中上传的单文件是公告信息文件，主要使用了 mkdir()和 move_uploaded_file()函数，下面分别对其进行介绍。

（1）mkdir()函数

该函数用来判断某文件是否存在，并且是否可写，如果满足上述条件则返回 true，否则返回 false。该函数的语法如下：

```
bool mkdir(string pathname [, int mode])
```

mkdir()函数的参数说明如表 5.1 所示。

表 5.1　mkdir()函数的参数说明

参　　数	说　　明
pathname	必要参数。用于指定新建目录的名称
mode	可选参数。用于指定新建目录的模式

（2）move_uploaded_file()函数

该函数应用 POST 方法实现文件的上传。其语法如下：

```
bool move_uploaded_file(string filename, string destination)
```

move_uploaded_file()函数的参数说明如表 5.2 所示。

表 5.2　move_uploaded_file()函数的参数说明

参　　数	说　　明
filename	必要参数。指定要上传的文件地址
destination	必要参数。上传到服务器后的存储目录及名称

注意　应用 POST 方法上传文件时，必须在上传表单的<form>标记中添加属性“enctype="multipart/form-data"”。

5.2.4　多图片上传

本模块在上传公告信息图片时，采用多图片上传方式，使用 for 循环语句循环获取图片信息，然后再使用 move_upload_file()函数执行上传操作。

for 语句是 PHP 中最复杂的循环控制语句，拥有 3 个条件表达式。其语法如下：

```
for (expr1; expr2; expr3){
    statement
}
```

for 循环语句的参数说明如表 5.3 所示。

表 5.3　for 循环语句的参数说明

参　　数	说　　明
expr1	必要参数。第 1 个条件表达式，在第 1 次循环开始时被执行
expr2	必要参数。第 2 个条件表达式，在每次循环开始时被执行，决定循环是否继续
expr3	必要参数。第 3 个条件表达式，在每次循环结束时被执行
statement	必要参数。满足条件后，循环执行的语句

其执行的过程是：首先执行表达式 1，然后执行表达式 2，并对表达式 2 的值进行判断，如果值为真，则执行 for 循环语句中指定的内嵌语句（如果值为假，则直接结束循环，跳出 for 循环语句），最后执行表达式 3（切记是在表达式 2 的值为真时），并返回表达式 2 继续循环执行。for 循环语句的操作流程如图 5.8 所示。

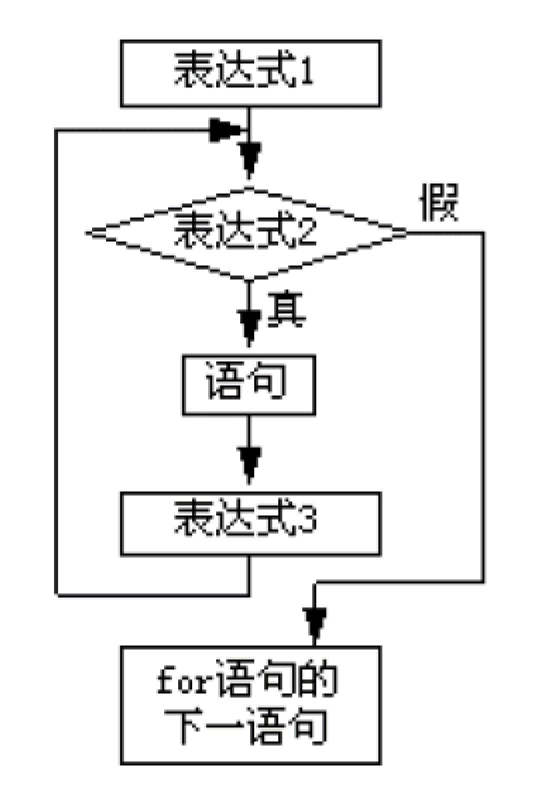


图 5.8　for 循环语句的流程图

5.2.5　读取指定目录中的目录及文件

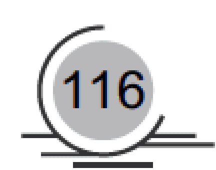
读取指定目录中的目录及文件主要使用 opendir()、getcwd()和 iconv()函数，以及 while 循环语句，循环输出目录文件。

（1）opendir()函数

该函数用来打开指定的目录，并返回该目录的标识。其语法如下：

```
resource opendir(string path)
```

参数 path 为该函数所要打开目录的路径。





（2）getcwd()函数

该函数返回当前的工作目录。其语法如下：

```
string getcwd(void)
```

该函数的参数为空。

（3）iconv()函数

该函数将一种已知的字符编码格式 A 转换成另外一种编码格式 B。其语法如下：

```
string iconv( string in_charset, string out_charset, string str )
```

参数说明如下。

☑　in_charset：一种已知将被转换的编码格式 A。

☑　out_charset：另一种将要转换成的编码格式 B。

☑　str：指定被转换的字符串。该函数转换成功返回字符串，否则返回 false。

（4）while 循环语句

while 循环语句的作用是反复地执行某一项操作，是循环控制语句中最简单、最常用的一种。while 循环语句对表达式的值进行判断，当表达式为非 0 值时，执行 while 语句中的内嵌语句；当表达式的值为 0 时，则不执行 while 语句中的内嵌语句。该语句的特点是：先判断表达式，后执行语句。while 循环控制语句的操作流程如图 5.9 所示。

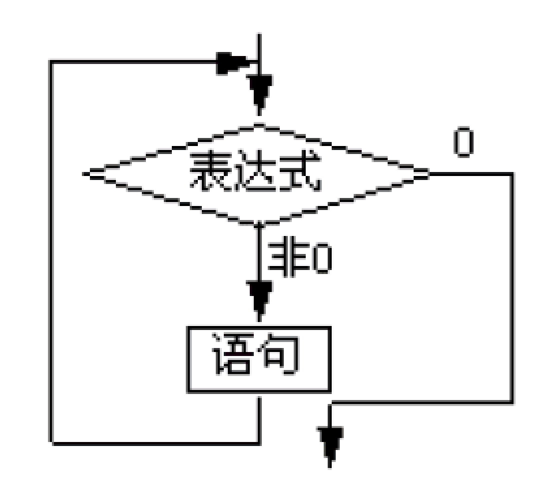


图 5.9　while 循环控制语句的操作流程

其语法如下：

```
while (expr){
        statement;                              /*先判断条件，当条件满足时执行语句块，否则
                                                 不向下执行*/
}
```

只要 while 表达式 expr 的值为 true，就重复执行嵌套中的 statement 语句，如果 while 表达式的值一开始就是 false，则循环语句一次也不执行。

5.2.6　使用 jQuery 控制弹出公告信息

假日公告管理系统中，控制公告信息的输出应用的是 jQuery 技术，下面对具体应用的方法进行介绍。

（1）$()，用于替代 document.getElementById()。语法如下：

```
var someElement = $("#myId");
```

例如，通过 jQuery 获取 DOM 对象中的元素，其代码如下：

```
$("div m");
$("div.admin");
```

```
$("div #mysqlid ");
$("table a",content);
```

第一行代码获取所有标签下的 m 元素。第二行代码获取 class 为 admin 的元素。第三行代码获取标签下面 id 为 mysqlid 的元素。第四行代码获取 content 为上下文的 table 里面所有的连接元素。

（2）slideToggle(speed,[callback])，通过高度变化来切换所有匹配元素的可见性，并在切换完成后可选地触发一个回调函数。

这个动画效果只调整元素的高度，可以使匹配的元素以滑动的方式隐藏或显示。

☑ 参数 speed：3 种预定速度之一的字符串（"slow"、"normal"或"fast"）或表示动画时长的毫秒数值（如 1000）。

☑ 可选参数 callback：在动画完成时执行的函数。

例如，用 600 毫秒缓慢地将段落滑上或滑下 。

```
$("p").slideToggle("slow");
```

例如，用 200 毫秒快速地将段落滑上或滑下，之后弹出一个对话框。

```
$("p").slideToggle("fast",function(){
    alert("Animation Done.");
 });
```

（3）ready(fn)，当 DOM 载入就绪可以查询及操纵时绑定一个要执行的函数。该方法是 window.load 事件的替代方法。语法如下：

```
ready(fn)
```

参数 fn 是在 DOM 就绪时执行的函数，可以为该参数任意起一个名字，并因此可以不再担心命名冲突而放心地使用$别名。

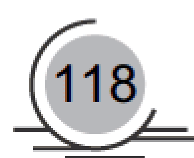
> **说明** 在使用 ready(fn)时，必须确保在<body>元素的 onload 事件中没有注册函数，否则不会触发$(document).ready()事件。可以在同一个页面中无限次地使用$(document).ready()事件。其中，注册的函数会按照代码中的先后顺序依次执行。

例如，通过$(document).ready(fn)代替<body>标签的 onload 事件，其代码如下：

```
$(document).ready(function(){
      alert("jQuery");
});
```

使用$(document).ready()的简写，同时内部的 jQuery 代码依然使用$作为别名，而不必管全局的$是什么。其 jQuery 代码如下：

```
jQuery(function($) {
  …//可以在这里继续使用$作为别名
});
```





上述$(document).ready()的简写替换的是如下代码：

```
<body onload="alert('jQuery');">
```

（4）css(name)，访问第一个匹配元素的样式属性。返回值为 string，参数 name 为要访问的属性名称。例如，取得第一个段落的 color 样式属性的值，其代码如下：

```
$("p").css("color");
```

（5）css(name,value)，在所有匹配的元素中，设置一个样式属性的值。数字将自动转化为像素值。参数 name 为属性名，参数 value 是属性值。

例如，初始状态下隐藏下拉列表，其代码如下

```
$("#cs").css("display","none");        //初始状态使城市下拉列表不可见
```

在假日公告管理系统中，需要在输出公告信息的网页中载入 jQuery 库文件，编写 JavaScript 脚本，控制公告信息的输出。关键代码如下：

```
<script src="js/jquery.js"></script>
<script>
    function showLayerNotice(){
        $("#Layer110").slideToggle(1000);
    }
    $(document).ready(function(){
        $("#Layer110").css("left", (screen.width-520)/2);
        setTimeout("showLayerNotice()", 1300) ;
    });
</script>
```

同时还要创建<div>标签，输出公告信息。在输出公告信息之前，要读取数据库中的数据，判断是否输出公告信息。如果指定字段值为 1，则包含上传的.html 文件，即输出公告信息，否则不输出<div>标签中的内容。关键代码如下：

```
<?php
include("conn/conn.php");                    //连接数据库
$result=mysql_query("select perform from tb_01",$conn);//执行查询语句
$myrow=mysql_fetch_array($result);           //获取查询结果
if($myrow['perform']==1){                    //判断指定字段的值，如果等于 1 则输出公告信息，否则不输出
?>
<div id="Layer110"
    style="position: absolute; border:3px solid #CCCCCC; left: 50px; top: 50px; z-index: 1; font-size: 0px; display: none">
<?php
    include("upfile/index.html");            //包含公告文件
?>
</div>
<?php
}
?>
```

5.3　数据库设计

5.3.1　数据库分析

根据系统的功能结构找出需要保存的信息（也可以理解为现实世界中的实体），并将其转化为原始数据（属性类型）形式。这种描述现实世界的概念模型可以使用 E-R 图来表示，也就是实体 E-R 图。最后，将 E-R 图转换为关系数据库。这里重点介绍两个 E-R 图。

1．假日公告信息实体

假日公告信息实体包括 id、公告名称和是否显示 3 个属性，其 E-R 图如图 5.10 所示。

2．管理员信息实体

管理员信息实体包括 id、用户名和密码 3 个属性，其 E-R 图如图 5.11 所示。

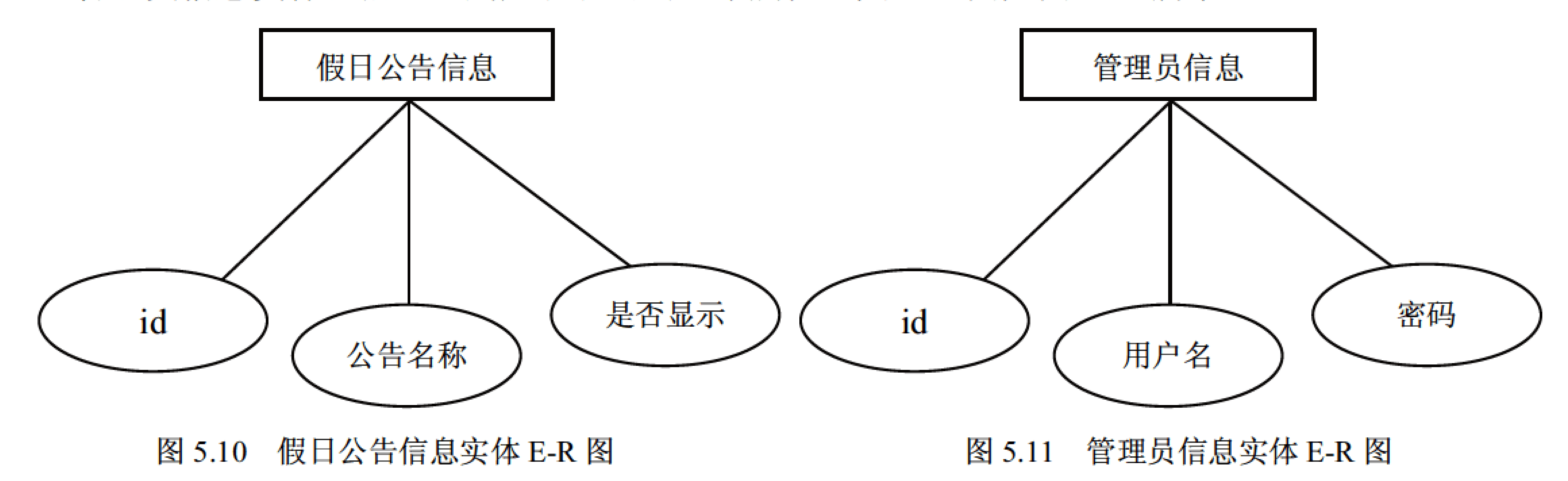


图 5.10　假日公告信息实体 E-R 图　　图 5.11　管理员信息实体 E-R 图

5.3.2　创建数据库和数据表

系统 E-R 图设计完成后，根据 E-R 图来创建数据库及数据表，先来看一下所使用的数据表情况，如图 5.12 所示。

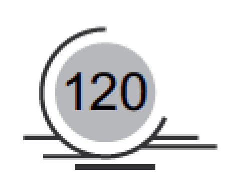
表	操作	记录数	类型	整理
tb_01		1	MyISAM	utf8_general_ci
tb_user		1	MyISAM	utf8_general_ci
2 个表	总计	2	MyISAM	utf8_general_ci

图 5.12　假日公告管理模块数据表

各个表的结构和说明介绍如下。

☑　tb_01（假日公告信息表）

假日公告信息表用于存储发布的假日信息，其结构如图 5.13 所示。




服务器: localhost ▸ 数据库: db_jiari ▸ 表 : tb_01

浏览 | 结构 | SQL | 搜索 | 插入 | 导出 | Import | 操作 | 清空

	字段	类型	整理	属性	Null	默认	额外		
	id	int(4)			否		auto_increment		
	name	varchar(50)	utf8_unicode_ci		否				
	perform	varchar(1)	utf8_general_ci		否				

图 5.13　假日公告信息表

☑　tb_user（管理员信息表）

管理员信息表主要用于存储管理员信息，其结构如图 5.14 所示。


服务器: localhost ▸ 数据库: db_jiari ▸ 表 : tb_user

浏览 | 结构 | SQL | 搜索 | 插入 | 导出 | Import | 操作 | 清空

	字段	类型	整理	属性	Null	默认	额外		
	id	int(4)			否		auto_increment		
	name	varchar(10)	utf8_unicode_ci		否				
	pwd	varchar(10)	utf8_unicode_ci		否				

图 5.14　管理员信息表

5.4　用户登录

5.4.1　用户登录概述

用户登录模块是假日公告管理系统的门户，只有登录成功后，才可以进入系统主页面，并进行发布、取消及删除假日信息等操作。用户登录模块运行结果如图 5.15 所示。



图 5.15　用户登录模块运行结果

5.4.2　用户登录功能实现过程

当用户输入登录信息后，首先判断用户名和密码是否为空，然后再判断用户名和密码是否正确，

如果正确则登录成功，进入系统主页面，否则将给出提示信息。

（1）判断用户名和密码是否为空，应用的是 checkit()自定义脚本函数，存储于 user.php 文件中。关键代码如下：

```
<script language="javascript">
function checkit(){                              //自定义函数
        if(form1.name.value==""){                //判断用户名是否为空
                alert("请输入用户名!");
                form1.name.select();
                return false;
        }
        if(form1.pwd.value==""){                 //判断密码是否为空
                alert("请输入密码!");
                form1.pwd.select();
                return false ;
        }
                return true;

}
</script>
```

（2）判断用户名和密码是否正确，如果正确则登录成功，否则将提示用户名和密码不正确，其操作在 user.php 文件中完成。代码如下：

```
<?php
include("conn/conn.php");                                  //包含数据库连接文件
if(isset($_POST['name']) and $_POST['pwd']!=null){  //判断用户名和密码是否为空
      $select=mysql_query("select * from tb_user where name='".$_POST['name']."' and pwd='".$_POST
['pwd']."'",$conn);                                        //执行查询操作
      if($row=mysql_num_rows($select)==1){
            $_SESSION['name']=$_POST['name'];
            echo "<script>alert('登录成功！');window.location.href='indexs.php';</script>;";
      }else{
            echo "<script>alert('用户名和密码不正确！');window.location.href='user.php';</script>;";
      }
}
?>
…                                                          //省略部分代码
<input type="image" name="imageField" onclick="return checkit();" src="images/user_05.gif" />
…                                                          //省略部分代码
<input type="image" name="imageField2" onclick"return false;" src="images/user_07.gif" />
```

5.5 假日公告发布

5.5.1 假日公告发布概述

假日公告发布模块用于发布假日公告信息。进入假日公告发布页面，填写假日名称，并选择假日



文件及图片，然后单击“发布”按钮，即可完成公告信息发布。假日公告信息发布页面运行效果如图 5.16 所示。



图 5.16　假日公告信息发布页面运行效果

5.5.2　假日公告信息发布实现过程

（1）创建 form 表单，以 POST 方式将数据提交到 promulgation_ok.php 文件中进行处理，并且设置“enctype="multipart/form-data"”属性，同时添加假日名称、假日文件、假日图片和是否执行等表单元素。关键代码如下：

```
<form action="promulgation_ok.php" method="post" enctype="multipart/form-data" name="form1" id="form1">
    <table id="__01" width="800" height="600" border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td colspan="3"><a href="indexs.php"><img src="images/manage_01.gif" alt="" width="800" height=
"201" border="0" /></a></td>
      </tr>
      <tr>
        <td><img src="images/manage_02.gif" width="140" height="251" alt="" /></td>
        <td width="523" height="251" align="center" valign="top" background="images/manage_03.gif"><table
width="416" height="232">
          <tr>
            <td width="89" height="30">假日名称：</td>
            <td width="315" align="left"><input name="name" type="text" size="40" /></td>
          </tr>
          <tr>
            <td height="30">假日文件：</td>
            <td align="left"><input name="files" type="file" size="31" /></td>
          </tr>
```

```
          <tr>
            <td height="30">假日图片 1：</td>
            <td align="left"><?php
         for($a=0;$a<4;$a++){
      ?>
                  <input type="file" name="photo_<?php echo $a;?>" />
                  <?php
         }
      ?>
            </td>
         </tr>
          <tr>
            <td height="30">是否执行：</td>
            <td><label>
              <input name="radio" type="radio" value="1" checked="checked" />
              是
              <input type="radio" name="radio" value="0" />
              否</label></td>
          </tr>
          <tr>
            <td height="30" colspan="2"><div align="center">
                  <input type="submit" name="Submit" value="发布" />
                  <input type="reset" name="Submit2" value="取消" />
            </div></td>
          </tr>
        </table></td>
        <td><img src="images/manage_04.gif" width="137" height="251" alt="" /></td>
      </tr>
      <tr>
        <td colspan="3"><img src="images/manage_05.gif" width="800" height="148" alt="" /></td>
      </tr>
    </table>
  </form>
```

（2）创建 promulgation_ok.php 文件，获取表单中提交的数据，完成假日公告信息的发布操作。关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_POST['name']) and $_POST['Submit']=="发布"){
     $insert=mysql_query("insert into tb_01(name,perform) values('".$_POST['name']."','".$_POST['radio']."')",
$conn);                                                        //添加数据
     if(isset($_FILES['files']['name'])){                      //判断文件及图片是否存在
          if(!is_dir("./upfile/images")){                      //判断指定目录是否存在
               mkdir("./upfile/images");                       //创建目录
          }
          $photo="upfile/".$_FILES['files']['name'];           //文件的存储路径和名称
          if(move_uploaded_file($_FILES['files']['tmp_name'],$photo)){   //执行上传
```



```
                    for($a=0;$a<4;$a++){                                    //循环语句
                        $name=$_FILES['photo_'.$a];                         //将变量的名称保存在变量中
                        $path='upfile/images/'.$name['name'];               //定义上传文件的路径
                        move_uploaded_file($name['tmp_name'],$path);
                    }
                    echo "<script>alert('上传成功！');window.location.href='promulgation.php';</script>";
                }else{
                    echo "<script>alert('上传失败！');window.location.href='promulgation.php';</script>";
                }
        }
}else{
        echo "<script>alert('请选择上传文件！');window.location.href='promulgation.php';</script>";
}
?>
```

5.6　假日公告取消

5.6.1　假日公告取消概述

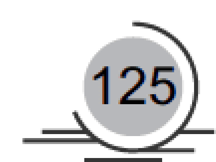
假期过后，在假日取消页面中单击假日名称后的“取消”超链接，即可取消该假日公告，并在数据库中删除该假日信息。假日公告信息取消页面运行效果如图 5.17 所示。



图 5.17　假日公告信息取消页面运行效果

5.6.2　浏览公告信息

打开假日信息取消页面，默认将显示所有假日信息。通过 select 查询语句，查询假日公告信息表



中的数据，通过 while 语句循环输出假日公告信息，并且为每条信息设置一个“取消”超链接，将信息的 id 作为参数值传递到 update.php 文件中。cancel.php 的关键代码如下：

```
<?php
include("conn/conn.php");
$select=mysql_query("select * from tb_01",$conn);
while($array=mysql_fetch_array($select)){
?>
<tr>
      <td height="30"><?php echo $array['name'];?> </td>
        <td><div align="center">
             <a href="update.php?abrogation=<?php echo $array['id'];?>">取消</a>
      </div></td>
/tr>
<?php
}
?>
```

5.6.3　取消假日公告

取消假日公告主要是通过超链接参数 abrogation 传递的值，在 update.php 页中执行 delete 删除语句删除信息，取消假日公告。update.php 的关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
include("conn/conn.php");
if(isset($_GET['abrogation'])){
      $update=mysql_query("delete from tb_01 where id='".$_GET['abrogation']."'",$conn);
      if($update){
             echo "<script>alert('假日信息取消成功！');window.location.href='cancel.php';</script>";
      }else{
             echo "<script>alert('假日信息取消失败！');window.location.href='cancel.php';</script>";
      }
}
?>
```

5.7　假日公告信息管理

5.7.1　假日公告信息管理概述

假日公告信息管理模块对上传的公告文件及公告图片进行管理，将过期的公告信息文件删除，以免造成空间浪费。假日公告信息管理模块运行效果如图 5.18 所示。

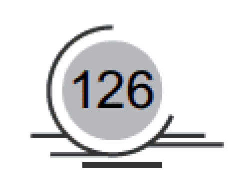






图 5.18　假日公告管理模块运行效果

5.7.2　假日公告信息管理实现过程

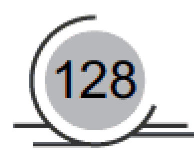
（1）创建 manage.php 动态页，输出指定目录下的文件夹及文件，并实现“上级目录”的跳转。关键代码如下：

```
<?php
if (! isset ( $_GET ['catalog'] ) || empty ( $_GET ['catalog'] ))
	$current_directory = getcwd (); //获得脚本目录
else
	$current_directory = $_GET ['catalog'];
chdir ( iconv ( "utf-8", "gb2312", $current_directory ) ); //改变当前目录
echo "<span class='STYLE4'>当前目录:</span><span class='STYLE2'>" . iconv ( "gb2312", "utf-8", getcwd () ) .
"</span><br>";
$ml = opendir ( iconv ( "utf-8", "gb2312", $current_directory ) ); //打开目录
while ( $gain_directory = readdir ( $ml ) ) { //循环读取目录中的文件夹及文件
	echo "<tr><td align='center' valign='middle'>";
	if (is_dir ( $gain_directory )) { //判断是目录
		if ($gain_directory == ".") {
			$catalog = getcwd (); //显示当前目录
			$catalog = iconv ( "gb2312", "utf-8", $catalog );
			echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>锁定</span>
</a>";
		} elseif ($gain_directory == "..") {
			$catalog = getcwd () . "\\..";  //上级目录
			$catalog = iconv ( "gb2312", "utf-8", $catalog );
			if($catalog==""){
				echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>上级目
录</span></a>";
			}else{
				echo "<a href=manage.php?catalog=".urlencode($catalog)."><span class='style1'>上级目
录</span></a>";
```

```
                }
            } else {
                $catalog = getcwd () . "\\$gain_directory"; //子目录
                $catalog = iconv ( "gb2312", "utf-8", $catalog );
                echo "<a href=manage.php?catalog=".urlencode($catalog).">" . iconv ( "gb2312", "utf-8", $gain_
directory ) . "</a>";
            }
        } else {
            $ext = substr ( $gain_directory, strrpos ( $gain_directory, "." ) );
            if (strtoupper ( $ext ) == ".html" || strtoupper ( $ext ) == ".gif" || strtoupper ( $ext ) == ".jpg" ) {
                $catalog = getcwd ();
                $catalog = iconv ( "gb2312", "utf-8", $catalog );
                echo    iconv ( "gb2312", "utf-8", $gain_directory ) ;
            } else {
                echo iconv ( "gb2312", "utf-8", $gain_directory );
            }
        }
        if (is_dir ( $gain_directory ))
            $file_size = "目录";
        else
        $file_size = filesize ( $gain_directory );
        echo "<td align='center' valign='middle'>$file_size</td>";
        $create_time = date ( "Y-m-d H:i:s", filectime ( $gain_directory ) );
        echo "<td align='center' valign='middle'>" . iconv ( "gb2312", "utf-8", $create_time ) . "</td>";
        $update_time = date ( "Y-m-d H:i:s", filemtime ( $gain_directory ) );
        echo "<td align='center' valign='middle'>$update_time</td>";
        echo "<td align='center' valign='middle'>";
        if ($gain_directory == ".") {
            $catalog = getcwd (); //显示当前目录
            echo "删除";
        } elseif ($gain_directory == "..") {
            $catalog = getcwd () . "\\.."; //上级目录
            echo "删除";
        } else {
            $catalog = getcwd () . "\\$gain_directory"; //子目录
            echo  "<a  href='delete.php?catalog=".urlencode($catalog)."&filename="  .  urlencode(getcwd  ())  .  "'
title='删除目录或者文件' >删除</a>";
        }
        echo "</td>";
        echo "</tr>";
}
closedir ( $ml );
?>
```

（2）创建 delete.php 页，当用户单击“删除”超链接时，在 delete.php 页面接收超链接传递的参数，完成删除操作。关键代码如下：

```
<?php
header("content-type:text/html;charset=utf-8");
```



```
$count=substr_count($_GET['catalog'],'.');
if($count>=1){
	if(unlink($_GET['catalog'])){
		echo "<script>alert('文件删除成功!'); window.location.href='manage.php?catalog=".urlencode($_GET
['filename'])."';</script>";
	}else{
		echo "<script>alert('文件删除失败！ '); history.back();</script>";
	}
}else{
	if(is_dir($_GET['catalog'])){
		if(@rmdir($_GET['catalog'])){
			echo "<script>alert('目录删除成功！ '); window.location.href='manage.php?catalog=".urlencode
($_GET['filename'])."';</script>";
		}else{
			echo "<script>alert('目录删除失败！ '); history.back();</script>";
		}
	}
}
?>
```

说明 在本模块中，创建 indexs.php 文件，模拟网站主页，当完成假日公告信息的添加之后，运行 indexs.php 文件，将弹出公告信息。其运行效果如图 5.19 所示。



图 5.19　弹出公告信息效果

5.8　本 章 小 结

本章通过一个完整的假日公告管理系统带领读者详细讲解一个模块的开发流程。同时，在假日公告管理系统中，采用 jQuery 控制弹出公告信息，以及单文件上传和多图片上传的功能，使页面显示得更加有条理性。通过本章的学习，读者不仅可以了解一些网站的基本流程，而且可以熟悉添加和删除公告信息的实现方法。

第 6 章

分页显示模块

（PHP+Ajax+Smarty+PDO 实现）

在网站中，当面对数据库中的大量数据要输出时，在头脑中闪过的第一个想法就是应用什么分页技术输出数据库中的数据才更加合理、简单、快捷。分页功能是每个网站都必不可少的，在本模块中，将介绍几种分页的实现方法，希望对读者以后程序的开发有所帮助。

通过阅读本章，可以学习到：

- 自定义函数替换超长文本中的特殊字符
- 自定义函数控制超长文本的输出
- PDO 操作 MySQL 数据库
- PHP 超长文本分页
- Ajax 无刷新分页
- 分页类分页
- PHP 上下分页
- PHP 跳转分页



6.1　分页显示概述

6.1.1　模块概述

分页显示模块是网站开发中最常用的一个典型模块，该模块的最大用途就是将海量数据或者超长文本数据分页输出。这样一来，不仅使页面的整体效果更加紧凑，而且能带来最好的用户体验。

6.1.2　功能结构

在本模块中，整体上分为两部分内容：一部分是面向对象的分页；另一部分是面向过程的分页。在面向对象分页中，通过 Smarty 模板完成网页的动静分离，通过 PDO 操作 MySQL 数据库，并且将 Smarty 模板的配置、PDO 连接操作数据库以及分页方法都封装到类中，通过类中方法的调用完成各种操作；在面向过程的分页中，完成超长文本的分页、Ajax 无刷新分页、跳转分页和上下分页。本模块的系统功能结构如图 6.1 所示。

分页显示模块
- 面向对象分页
  - 模板配置类
  - 数据库连接类
  - 数据库操作类
  - 分页类
  - 类的实例化
- 面向过程分页
  - 超长文本分页类
  - 无刷新分页类
  - 跳转分页
  - 上下分页

图 6.1　分页显示模块功能结构图

6.1.3　程序预览

在分页显示模块中，展示了 5 种不同的分页方法：超长文本的分页显示、Ajax 无刷新分页、跳转分页、上下分页和通过分页类分页。

☑ 超长文本的分页显示：实现对文本文件中数据的分页显示，其主要应用在注册须知、公司简介、版权说明等文本类的文件输出中，如图 6.2 所示。

☑ Ajax 无刷新分页：实现数据库中数据的无刷新显示，其主要应用在一些支持视频播放的网站中，通过 Ajax 无刷新进行分页，不会影响到视频文件的播放，如图 6.3 所示。



图 6.2　PHP 超长文本分页显示的运行结果



图 6.3　Ajax 无刷新分页的应用

☑ 跳转分页：实现在当前页面中到其他指定页面的跳转，通过它与标准分页功能的结合使用可

以增加分页操作的灵活性。如图 6.4 所示。



图 6.4　PHP 跳转分页的运行结果

☑ 上下分页：针对当前的页面，实现上一页和后一页的跳转，也属于一种常用的分页方法，如图 6.5 所示。

☑ 通过分页类分页：将一种定义好的分页方法封装到一个类中，当需要进行分页操作时，可以直接调用这个类文件，并设置相应的参数，即可实现分页的操作，无须再重新编写分页方法。这是一个比较实用的分页方法，如图 6.6 所示。



图 6.5　PHP 上下分页的运行结果



图 6.6　分页类分页的运行结果

6.2　面向对象分页

通过分页类实现分页功能是一个非常好的方法。一个分页类创建成功后，就可以直接调用类文件实现分页功能，而无须再重新编译。在分页类模块中，采用 Smarty 模板完成网页的动静分离，通过 PDO 完成对数据库的连接和操作。分页类实现分页的运行结果如图 6.6 所示。

6.2.1　Smarty 模板的安装和配置

将 Smarty 模板的压缩包解压，复制 libs 目录到服务器 MR\06\01\system 目录下。





配置 Smarty 模板，在确定 Smarty 模板文件的位置之后，在 smarty 目录下新建 templates、templates_c、configs 和 cache 4 个目录，用于存储不同的文件。

封装 Smarty 模板的配置类，完成 Smarty 模板的配置操作，该类存储于 06\01\system\system.smarty.inc.php 文件中。其代码如下：

```
<?php
require("libs/Smarty.class.php");                //包含模板文件
class SmartyProject extends   Smarty{            //定义类，继承模板类
function SmartyProject(){                        //定义方法
$this->template_dir = "./";                      //指定模板文件存储位置
$this->compile_dir = "./system/templates_c/";    //指定编译文件存储位置
$this->config_dir = "./system/configs/";         //指定配置文件存储位置
$this->cache_dir = "./system/cache/";            //指定缓存文件存储位置
}
}
?>
```

6.2.2　加载 PDO 数据库抽象层

在本模块中，通过 PDO 数据库抽象层连接和操作 MySQL 数据库。PDO 是与 PHP 5.1 一起发行的，默认包含在 PHP 5.1 中。由于 PDO 需要 PHP 5 核心面向对象特性的支持，因此其无法在 PHP 5.0 之前的版本中使用。

默认情况下，PDO 在 PHP 5.2 中为开启状态，但是要启用对某个数据库驱动程序的支持，仍需要进行相应的配置操作。

在 Linux 环境下，要使用 MySQL 数据库，可以在 configure 命令中添加如下选项：

```
--with-pdo-mysql=/path/to/mysql/installation
```

在 Windows 环境下，PDO 在 php.ini 文件中进行配置，如图 6.7 所示。

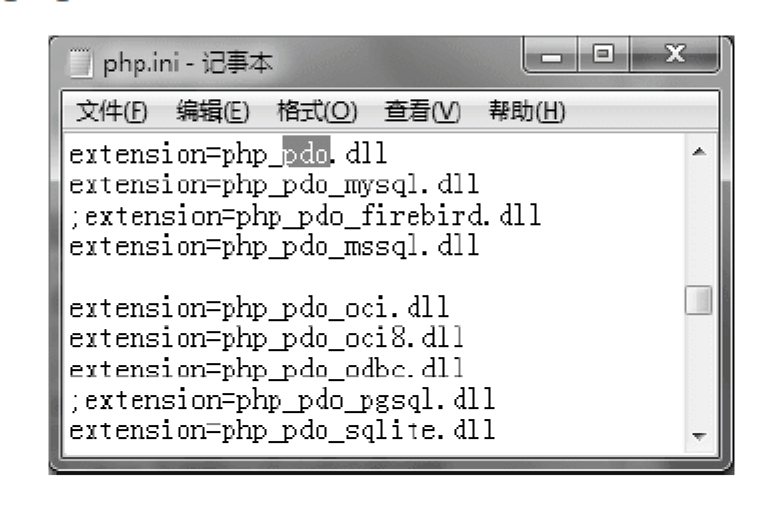


图 6.7　Windows 环境下配置 PDO

要启用 PDO，首先必须加载"extension=php_pdo.dll"，如果要想其支持某个具体的数据库，那么还要加载对应的数据库选项。例如，要支持 MySQL 数据库，则需要加载"extension=php_pdo_mysql.dll"选项。

> **注意**　在完成数据库的加载后，要保存 php.ini 文件，并且重新启动 Apache 服务器，修改才能够生效。有关 PDO 连接和操作 MySQL 数据库的方法将在技术提炼中做详细讲解，这里不再赘述。

6.2.3 分页类模块的页面设计

由于本模块的开发使用的是 Smarty 模板技术，因此模块的页面设计和程序的开发是完全分离的，模块的页面设计是在 index.tpl 文件中完成的，而程序的开发则是在 index.php 文件中完成。

分页类模块的页面设计主要就是将程序开发中模板变量中的数据进行输出，以及分页功能的实现。关键代码如下：

```
{if $isbbs=="T"}<!--首先通过模板变量$isbbs 判断是否有数据-->
<!--通过 section 内建函数，循环输出数组模板变量$arraybbs 中的图书数据-->
{php} $i=1;   {/php}
{section name=bbsid loop=$arraybbs}
<tr>
<td height="22" class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";}{/php}> 
<a href="index.php?id={$arraybbs[bbsid].tb_book_id}" target="_blank">
{unhtml content=$arraybbs[bbsid]. tb_book_name}</a>
</td>
<td class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";}{/php}>
<div align="center">{$arraybbs [bbsid].tb_book_synopsis}</div></td>
<td height="22" class="STYLE4"   {php} if($i%2==0){ echo"bgcolor='#F5F5F5'";}{/php}>
<div align="center"> {$arraybbs[bbsid].tb_print_date}</div></td>
<td height="22" class="STYLE4" {php} if($i%2==0){ echo "bgcolor='#F5F5F5'";}{/php}>
<div align="center"> {$arraybbs[bbsid].tb_book_author}</div></td>
</tr>
{php} $i++;   {/php}
{/section}
<table width="100%" height="22" border="0" align="center" cellpadding="0" cellspacing="0">
<tr><!--输出分页功能的信息-->
<td class="STYLE4"> {$showpage}</td>
</tr>
</table>
{/if}
<!--判断当没有数据时，则输出如下内容-->
{if $isbbstell=="F" && $isbbs=="F"}
<table width="96%" height="30" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#58BAE9">
<tr>
<td bgcolor="#FFFFFF"><div align="center">
<font color="#FF0000">{ldelim}该分类暂无论坛公告或帖子{rdelim}</font></div></td>
</tr>
</table>
{/if}
```

6.2.4 分页类模块的程序开发

分页类模块的程序开发是在 index.php 文件中进行的，是一个纯粹的 PHP 代码页。在该文件中调用类文件，实现与数据库的连接，读取数据库中的数据；调用分页类，实现数据的分页显示，并且定义模板变量，用于在模板页中输出数据。关键代码如下：





```
<?php
header("Content-Type:text/html;charset=gb2312");
require_once("system/system.inc.php");   //调用指定的文件
$arraybbstell=$admindb->ExecSQL("select * from tb_mr_book",$conn);          //执行 select 查询语句
if(!$arraybbstell){
$smarty->assign("isbbstell","F");             //判断如果执行失败，则输出模板变量 isbbstell 的值为 F
}else{
$smarty->assign("isbbstell","T");             //判断如果执行成功，则输出模板变量 isbbstell 的值为 T
$smarty->assign("arraybbstell",$arraybbstell);          //定义模板变量 arraybbstell，输出数据库中的数据
}
if(isset($_GET["page"])){
$page=$_GET["page"];
}else{
$page=1;
}
$arraybbs=$seppage->ShowData("select * from tb_mr_book",$conn,1,$page); //调用分页类，实现分页功能
if(!$arraybbs){
$smarty->assign("isbbs","F");
}else{
$smarty->assign("isbbs","T");
$smarty->assign("showpage",$seppage->ShowPage("","","","","")); //定义输出分页数据的模板变量 ShowPage
$smarty->assign("arraybbs",$arraybbs);
}
$smarty->display("index.tpl");          //指定连接的模板页
?>
```

至此，分页类模块的讲解完毕，程序的完整代码请参考本章光盘中的内容。

6.3　超长文本分页

6.3.1　超长文本分页功能概述

通过 PHP 超长文本的分页显示功能可以对指定文本文件中的数据进行分页输出。运行结果如图 6.8 所示。



图 6.8　PHP 超长文本分页显示的运行结果

6.3.2　超长文本分页功能实现过程

PHP 超长文本的分页显示主要应用的是自定义函数 unhtml()和 msubstr()，有关这两个自定义函数的创建请参考 6.8.1 节和 6.8.2 节中的内容，这里不再赘述。

PHP 超长文本分页显示的操作在 06\mr_copyright.php 文件中完成。首先，判断变量的值是否为空，并调用指定的包含文件，获取自定义函数。代码如下：

```
if(isset($_POST['page'])){
$page=$_POST['page'];
}else if(isset($_GET['page'])){
$page=$_GET['page'];
}else{
$page=1;
}
include("function.php");                                    //包含自定义函数
```

然后，通过 file_get_contents()函数获取 file.txt 文本文件中的数据，返回到一个字符串中；通过 unhtml()函数替换字符串中的特殊字符；通过 strlen()函数获取字符串的长度；通过 ceil()函数计算将文本文件中的数据分成几页进行输出；通过 msubstr()函数控制数据的输出；应用 substr()函数获取到当前页输出的内容。代码如下：

```
<?php
if($page){
$counter=file_get_contents("file/file.txt");                //读取文本文件中的数据，返回一个字符串
$length=strlen(unhtml($counter));                           //计算字符串的长度，替换特殊字符
$page_count=ceil($length/1050);                             //对字符串进行分页处理
$c=msubstr($counter,0,($page-1)*1050);                      //通过自定义函数获取上一页中的数据
$c1=msubstr($counter,0,$page*1050);                         //通过自定义函数获取当前页中的数据
echo substr($c1,strlen($c),strlen($c1)-strlen($c));         //获取到当前页中要输出的数据
}
?>
```

最后，创建分页超链接，实现首页、上一页、下一页和尾页之间的跳转。代码如下：

```
<td width="42%" align="center" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  页次：<?php echo $page;?> / <?php echo $page_count;?>页</span></td>
<td width="58%" height="28" align="left" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  分页：
<?php
if($page!=1){
echo   "<a href=index.php?lmbs=明日版权声明&page=1>首页</a> ";
echo "<a href=index.php?lmbs=明日版权声明&page=".($page-1).">上一页</a> ";
}
if($page<$page_count){
echo "<a href=index.php?lmbs=明日版权声明&page=".($page+1).">下一页</a> ";
echo   "<a href=index.php?lmbs=明日版权声明&page=".$page_count.">尾页</a>";
```

```
}
?>
</span> </td>
```

至此，PHP 超长文本的分页显示介绍完毕。

6.4　Ajax 无刷新分页

6.4.1　Ajax 无刷新分页功能概述

Ajax 无刷新分页的强大功能体现最为明显的地方就是在播客中。因为在播客中，在播放视频文件的同时，如果使用的不是无刷新分页，那么当执行翻页的操作后，视频文件将被重新打开；如果使用的是无刷新分页，就不会出现这种情况。

为了更好地理解 Ajax 无刷新分页的强大功能，这里创建了一个视频文件，并且应用 Ajax 无刷新分页输出文本文件中的内容，验证无刷新分页的操作是否对视频文件的播放有影响。运行结果如图 6.9 所示。



图 6.9　Ajax 无刷新分页的应用

6.4.2　Ajax 无刷新分页功能实现过程

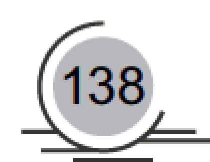
Ajax 无刷新分页由 mr_synopsis.php、mr_synopsis_OK.php 和 discuss_js.js 3 个文件组成。

在 mr_synopsis.php 文件中，首先，创建一个 div 标签，设置 div 标签的 id="synopsis"。然后，在 div 标签中编写超长文本分页输出的代码，并且在超长文本分页的超链接中应用 onclick 事件，调用自定义函数 artpagination()，传递的参数为另外一个实现分页功能的文件 mr_synopsis_ok.php。关键代码如下：

```
<div id="synopsis"><!--创建 div 标签，用于获取 js 文件中返回的分页结果-->
<table width="425" border="0" cellspacing="0" cellpadding="0">
```

```
<tr>
<td height="20"> </td>
</tr>
<tr>
<td>
<?php
//读取超长文本中的数据，实现超长文本中数据的分页显示
if($page){
$counter=file_get_contents("file/mr_synopsis.txt");
$length=strlen($counter);
$page_count=ceil($length/950);
$c=msubstr($counter,0,($page-1)*950);
$c1=msubstr($counter,0,$page*950);
echo substr($c1,strlen($c),strlen($c1)-strlen($c));
}
?>      </td>
</tr>
<tr>
<td>
<table border="0" align="center" cellpadding="0" cellspacing="0">
<tr><!--设置超长文本分页显示的超链接-->
<td width="42%" align="center" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  页次：<?php echo $page;?> / <?php echo $page_count;?>页</span></td>
<td width="58%" height="28" align="left" valign="middle" bgcolor="#FFFFFF">
<span class="STYLE1">  分页：
<?php
if($page!=1){
?>
<!--调用 artpagination()函数，实现无刷新的分页输出-->
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=1")'>首页</a> 
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?&page=<?php echo $page-1;?>")'>上一页</a>
 
<?php   }
if($page<$page_count){
?>
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=<?php echo $page+1;?>")'>下一页</a>
 
<a href="#" onclick='return artpagination("mr_synopsis_ok.php?page=<?php echo $page_count;?>")'>尾页</a>
<?php
}
?>
</span> </td>
</tr>
</table></td>
</tr>
</table>
</div>
```

最后，通过 js 文件中的自定义函数 artpagination()将 mr_synopsis_ok.php 文件中的分页结果返回到





div 标签 synopsis 中，实现无刷新分页的功能。js 文件中的关键代码如下：

```
//使用 XMLHttpRequest 对象创建异步 HTTP 请求
function artpagination(url){                        //创建自定义函数，获取传递的参数
xmlHttp.open('get',url,true);                        //通过 GET 方法，执行另外一个实现分页功能的文件
xmlHttp.onreadystatechange = function(){
if(xmlHttp.readystate == 4 && xmlHttp.status == 200){     //将结果返回到 div 标签 synopsis 中
document.getElementById("synopsis").innerHTML = xmlHttp.responseText;
}
}
xmlHttp.send(null);
}
```

说明 在 mr_synopsis_ok.php 文件中完成的分页功能与 mr_synopsis.php 文件中是相同的，代码不再重复。

6.5 标 准 分 页

6.5.1 PHP 标准分页功能概述

PHP 标准分页在当前页面中显示指定的记录，同时以 10 页为一个单位，实现上 10 页和下 10 页的跳转；并且实现当前页中显示的 10 个超链接的任意跳转。运行结果如图 6.10 所示。



图 6.10　PHP 标准分页运行结果

6.5.2 PHP 标准分页功能实现过程

在 PHP 标准分页中，每页显示 5 条数据。分页跳转可以分为 3 个部分：第 1 部分，根据查询结果，通过 while 语句在当前页面中显示 5 条记录信息；第 2 部分，以 10 页为一个单位，实现上 10 页和下

10 页的跳转；第 3 部分，在当前页中，在显示的 10 个超链接中任意跳转。PHP 标准分页由 mr_book.php 文件来完成，其具体操作步骤如下：

（1）连接数据库，定义分页中要使用的变量。包括数据库的连接文件 conn.php、当前页值的变量 $page 和当前 10 页值的变量$link_type。关键代码如下：

```
<?php
include_once("conn/conn.php");
if(isset($_POST['page'])){
$page=$_POST['page'];
}else if(isset($_GET['page'])){
$page=$_GET['page'];
}else{
$page=1;
}
if (isset($_GET['link_type'])) {
$link_type=$_GET['link_type'];
}else{
$link_type=0;
}
if (isset($_GET['vv'])) {
$vv=$_GET['vv'];
}else{
$vv=0;
}
include("function.php");
?>
```

说明 上述的这段代码存储于 index.php 文件中，因为本模块应用 switch 语句完成网页的框架操作，所有功能都包含于 index.php 文件中，所以将上述的这段代码存储于 index.php 文件中，作为公共内容。

（2）在 mr_book.php 文件中，从数据库中读取数据，统计出总的记录数，设置每页显示的记录数，计算出共有几页，并通过 while 语句循环输出数据库中的数据。代码如下：

```
<?php
if($page){                                          //判断当前页变量的值是否存在
$page_size=5;                                       //定义每页输出的记录数
$query="select * from tb_forum_send ";              //编写查询语句，应用 count 统计总的记录数
$result=$conn->query($query);                       //执行查询语句
$res=$result->fetchAll(PDO::FETCH_ASSOC);
$message_count=count($res);
$page_count=ceil($message_count/$page_size);        //计算出总共有几页
$offset=($page-1)*$page_size;                       //输出上一页结束的记录数
$query_2=$conn->query("select * from tb_forum_send limit $offset, $page_size");
foreach($query_2 as $value){
?>
<tr>
```

```
<td width="35%" align="center" bgcolor="#FFFFFF">
<a href="index_ok.php?send_id=<?php echo $value['tb_send_id'];?>" target="_blank">
<?php echo $value['tb_send_subject'];?></a>
</td>
<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_date'];?></td>
<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_user'];?></td>
<td width="10%" align="center" bgcolor="#FFFFFF">
<?php
$query_s=$conn->query("select * from tb_forum_restore ");
echo count($query_s);
?>
</td>
</tr>
<?php
}
}
?>
```

（3）创建分页超链接，实现分页中以 10 页为单位的跳转，以及当前 10 页中不同页面之间的跳转。关键代码如下：

```
<table width="800" height="30" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor=
"#D0D0D0">
<tr>
<td height="22" align="center" bgcolor="#FFFFFF">
<span class="STYLE2">帖子统计：<?php echo $message_count;?> 条  </span></td>
<td align="center" bgcolor="#FFFFFF">每页显示：<?php echo $page_size; ?>条</td>
<td align="center" bgcolor="#FFFFFF">共有：<?php echo $page_count;?>页</td>
<td align="center" bgcolor="#FFFFFF"><span class="STYLE2"> 分页：
<?php
$next=$link_type*10;            //以 10 页为一个单位
$n=$link_type-1;                //上一个 10 页的值
$m=$link_type+1;                //下一个 10 页的值
$prev_page=$page-10;            //当前页的上 10 页的值
$mm="";
if($link_type==0){              //判断变量的值是否等于 0，如果等于 0，则输出首页图标，无超链接
echo "<img src=\"images/02.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"首页\">";
}else{                          //否则，为首页图标设置超链接，连接到首页，并为上 10 页的图标设置超链接
echo "<a href='index.php?lmbs=明日论坛&vv=0&link_type=0&page=1'>
<img src=\"images/02.jpg\" width=\"8\" border=\"0\" height=\"9\" border=\"0\" title=\"首页\"></a> ";
$ccc=$vv-10;                    //定义变量，变量$vv 是当前 10 页的起始位置，$ccc 是上一个 10 页的起始位置
echo "<a href='index.php?lmbs=明日论坛&vv=$ccc&link_type=$n&page=$prev_page'>
<img src=\"images/01.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"上 10 页\"></a>";
}
?>
<?php
//通过 for 循环语句，输出当前页中的 10 个超链接
for($j=1;$j<=$page_count;$j++) {    //根据统计的总的记录数循环输出
$pnext=$next+$j;                //定义当前页中输出的页码值
```

```
if($mm==10){                    //判断当$mm 的值等于 10 时跳出循环
break;
}
if($mm>$page_count){            //判断当$mm 的值大于总记录时跳出循环
break;
}
if($page_count-$vv<10){
if($mm>=$page_count-$vv){       //判断当$mm 的值大于等于总记录减去当前 10 页的起始值时跳出循环
break;
}
}
echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$link_type&page=$pnext'>";
if($page==$pnext){              //为当前点击的页码描红
echo "<span class='STYLE1'> $pnext </span>";
}else{
echo " $pnext ";                //输出当前 10 页超链接中的数据
}
echo "</a>";
$mm=$mm+1;
}
?>
</span><span class="STYLE2">
<?php
$vv=$vv+$mm;                    //定义变量的值，定义下 10 页的起始位置
if($page_count-$vv<=0){         //判断当总记录数小于等于下 10 页的起始位置时，输出尾页的图标
echo "<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"尾页\">";
}else{                          //否则输出下 10 页的超链接
echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$m&page=$pnext'>
<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"下十页\"></a>";
}
if($message_count==0){          //判断当总记录数等于 0 时，输出没有记录
echo "没有记录!";
}
?>
</span>  </td>
</tr>
</table>
```

至此，有关 PHP 标准分页的功能介绍完毕，详细代码可以参考本章光盘中的内容。

6.6 跳 转 分 页

6.6.1 PHP 跳转分页功能概述

PHP 跳转分页的特点是能够实现当前页面到其他指定页面的跳转，多数情况下它与标准分页功能结合使用，以增加分页操作的灵活性。运行结果如图 6.11 所示。





图 6.11　PHP 跳转分页的运行结果

6.6.2　PHP 跳转分页功能的实现过程

在 PHP 跳转分页中，每页显示 5 条数据。分页跳转可以分为 3 个部分：第 1 部分，通过文本框实现指定页面的跳转功能；第 2 部分，以 10 页为一个单位，实现上 10 页和下 10 页的跳转；第 3 部分，在当前页中，在显示的 10 个超链接中任意跳转。PHP 跳转分页由 mr_forum.php 文件来完成，其具体操作步骤如下：

（1）连接数据库，定义分页中要使用的变量，包括数据库的连接文件 conn.php，当前页值的变量 $page 和当前 10 页值的变量$link_type。关键代码如下：

```
<?php
include_once("conn/conn.php");
if(isset($_POST['page'])){
$page=$_POST['page'];
}else if(isset($_GET['page'])){
$page=$_GET['page'];
}else{
$page=1;
}
if (isset($_GET['link_type'])) {
$link_type=$_GET['link_type'];
}else{
$link_type=0;
}
if (isset($_GET['vv'])) {
$vv=$_GET['vv'];
}else{
$vv=0;
}
include("function.php");
?>
```

说明 上述的这段代码存储于 index.php 文件中，因为本模块应用 switch 语句完成网页的框架操作，所有的功能都包含于 index.php 文件中，所以将上述的这段代码存储于 index.php 文件中，作为公共内容。

（2）在 mr_forum.php 文件中，从数据库中读取数据，统计出总的记录数，设置每页显示的记录数，计算出共有几页，并通过 while 语句循环输出数据库中的数据。代码如下：

```
<?php
if($page){                                    //判断当前页变量的值是否存在
$page_size=5;                                 //定义每页输出的记录数
$query="select * from tb_forum_send ";        //编写查询语句，应用 count 统计总的记录数
$result=$conn->query($query);                 //执行查询语句
$res=$result->fetchAll(PDO::FETCH_ASSOC);
$message_count=count($res);
$page_count=ceil($message_count/$page_size); //计算出总共有几页
$offset=($page-1)*$page_size;                 //输出上一页结束的记录数
$query_2=$conn->query("select * from tb_forum_send limit $offset, $page_size");
foreach($query_2 as $value){
?>
<tr>
<td width="35%" align="center" bgcolor="#FFFFFF">
<a href="index_ok.php?send_id=<?php echo $value['tb_send_id'];?>" target="_blank">
<?php echo $value['tb_send_subject'];?></a>
</td>
<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_date'];?></td>
<td width="25%" align="center" bgcolor="#FFFFFF"><?php echo $value['tb_send_user'];?></td>
<td width="10%" align="center" bgcolor="#FFFFFF">
<?php
$query_s=$conn->query("select * from tb_forum_restore ");
echo count($query_s);
?>
</td>
</tr>
<?php
}
}
?>
```

（3）创建一个表单，用于提交跳转的页码值，并且通过 JavaScript 脚本判断提交的值是否有效。代码如下：

```
<form name="form1" method="post" action="index.php?lmbs=明日论坛" onSubmit="return chk()">
<input type="submit" name="Submit" value="跳转"> 
<input name="page" type="text" size="3">
<input type="hidden" name="pages" value="<?php echo $page_count;?>">
</form>
```





判断提交的页码值是否有效是通过自定义函数 chk()完成的，其关键代码如下：

```
<script language="javascript">
function chk(){
if(form1.page.value<=0 || parseInt(form1.page.value)>parseInt(form1.pages.value)){
alert("您输入的页码无效!!");
form1.page.focus();
return(false);
}
return(true);
}
</script>
```

（4）编写 PHP 语句，实现分页中以 10 页为单位的跳转，以及当前 10 页中不同页面之间的跳转。关键代码如下：

```
<td align="center" bgcolor="#FFFFFF"><span class="STYLE2"> 分页：
<?php
$next=$link_type*10;            //以 10 页为一个单位
04    $n=$link_type-1;          //上一个 10 页的值
$m=$link_type+1;                //下一个 10 页的值
$prev_page=$page-10;            //当前页的上 10 页的值
$mm="";
if($link_type==0){              //判断变量的值是否等于 0，如果等于 0，则输出首页图标，无超链接
echo "<img src=\"images/02.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"首页\">";
}else{                          //否则，为首页图标设置超链接，连接到首页，并为上 10 页的图标设置超链接
echo "<a href='index.php?lmbs=明日论坛&vv=0&link_type=0&page=1'>
<img src=\"images/02.jpg\" width=\"8\" border=\"0\" height=\"9\" border=\"0\" title=\"首页\"></a> ";
$ccc=$vv-10;                    //定义变量，变量$vv 是当前 10 页的起始位置，$ccc 是上一个 10 页的起始位置
echo "<a href='index.php?lmbs=明日论坛&vv=$ccc&link_type=$n&page=$prev_page'>
<img src=\"images/01.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"上 10 页\"></a>";
}
?>
<?php
//通过 for 循环语句，输出当前页中的 10 个超链接
for($j=1;$j<=$page_count;$j++) {          //根据统计的总的记录数循环输出
$pnext=$next+$j;                          //定义当前页中输出的页码值
if($mm==10){                              //判断当$mm 的值等于 10 时跳出循环
break;
}
if($mm>$page_count){                      //判断当$mm 的值大于总记录时跳出循环
break;
}
if($page_count-$vv<10){
if($mm>=$page_count-$vv){                 //判断当$mm 的值大于等于总记录减去当前 10 页的起始值时跳出循环
break;
}
}
echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$link_type&page=$pnext'>";
if($page==$pnext){                   //为当前点击的页码描红
```

```
echo "<span class='STYLE1'> $pnext </span>";
}else{
echo " $pnext ";                    //输出当前 10 页超链接中的数据
}
echo "</a>";
$mm=$mm+1;
}
?>
</span><span class="STYLE2">
<?php
$vv=$vv+$mm;                        //定义变量的值，定义下 10 页的起始位置
if($page_count-$vv<=0){             //判断当总记录数小于等于下 10 页的起始位置时，输出尾页的图标
echo "<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"尾页\">";
}else{                              //否则输出下 10 页的超链接
echo "<a href='index.php?lmbs=明日论坛&vv=$vv&link_type=$m&page=$pnext'>
<img src=\"images/03.jpg\" width=\"8\" height=\"9\" border=\"0\" title=\"下十页\"></a>";
}
if($message_count==0){              //判断当总记录数等于 0 时，输出没有记录
echo "没有记录!";
}
?>
</span>  </td>
```

至此，有关 PHP 跳转分页的功能介绍完毕，详细代码可以参考本章光盘中的内容。

6.7 上 下 分 页

6.7.1 PHP 上下分页功能概述

PHP 上下分页功能以当前页码为基础，实现上一页和下一页的跳转，并且输出上一页中最后一条数据和下一页中第一条数据的内容。运行结果如图 6.12 所示。



图 6.12 PHP 上下分页的运行结果





6.7.2　PHP 上下分页功能的实现过程

PHP 上下分页，每页显示 2 条数据。分页跳转可以分成两个部分：第 1 部分是上下翻页的跳转，第 2 部分则是以 10 页为一个单位，实现上 10 页和下 10 页的跳转，以及在当前页显示的 10 个超链接中的任意跳转。有关第 2 部分的内容已经在第 6.6.2 节中做了详细介绍，这里不再赘述。

这里主要介绍上下翻页的跳转，以及输出上一页的最后一条数据和下一页的第一条数据的方法，还有分栏功能的实现方法。其操作在 mr_dictionary.php 文件中完成。

在输出数据库中的数据时，应用的是分栏方法，创建一个 1 行 2 列的表格来输出数据库中的数据。实现的原理如下：

（1）在 while 语句之外，定义一个变量$number=0。

（2）在 while 语句之中，通过 if 语句判断取模（$number % 2）的值是否等于 0，如果等于 0，则是行的开始（执行<tr>）。

（3）在 while 语句之中，通过 if 语句判断取模（$number % 2）的值是否等于 0，如果不等于 0，则是行的结束（执行</tr>）。

（4）在 while 语句之外，执行$number++。

关键代码如下：

```
<?php
$number=0;                                              //定义变量
//省略了部分 PHP 代码
$query_2=mysql_query("select * from tb_mr_bccd limit $offset, $page_size");
while($myrow_2=mysql_fetch_array($query_2)){           //循环输出
if (( $number % 2 ) == 0){                              //判断取模的值是否等于 0
?>
<tr>                                                    <!--当取模的值等于 0 时执行此操作-->
<?php     }      ?>
<td colspan="2" align="center"></td>                    <!--输出数据库中的数据-->
<?php
if ( ( $number %2 ) != 0){                              //判断当取模的值不等于 0 时
?>
</tr>                                                   <!--当取模的值不等于 0 时执行此操作-->
<?php }                                                 //结束 while 循环
$number++;                                              //变量值增加
?>
```

上一页、下一页跳转功能的实现，关键就是对当前页值 page 的判断。首先设置上一页的超链接和获取上一页中的最后一条数据，关键代码如下：

```
<td width="205" align="center" bgcolor="#FFFFFF">上一部：<a href="index.php?lmbs=明日编程词典&page=
<?php
if($page=="" or $page<=1){
echo 1;
}else{
echo ceil(($page*2-2)/2);
}
```

```
?>">
<?php
$page_3=($page-2)*2+1;
if($page_3<=0){
$page_3=0;
}
$query_3="select * from tb_mr_bccd limit $page_3,1";
$result3=$conn->prepare($query_3);                    //准备查询语句
$result3->execute();                                  //执行查询语句，并返回结果集
$res3=$result3->fetch(PDO::FETCH_ASSOC);
echo $res3['tb_bccd_name'];
?></a></td>
```

然后获取下一页超链接中 page 的值和下一页中的第一条数据，关键代码如下：

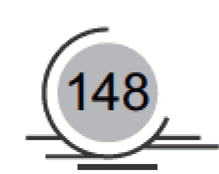
```
<td width="238" align="center" bgcolor="#FFFFFF"> 下 一 部 ： <a href="index.php?lmbs= 明 日 编 程 词 典
&page=<?php
if($page=="" or $page<=1){
echo 2;
}else if($page>=$page_count){
echo $page_count;
}else{
echo ceil(($page*2+1)/2);
}?>">
<?php
$page_4=($page)*2;
if($page_4<=0){
$page_4=2;
}
if($page_4==$page_count*2){
$page_4=$page_count*2-1;
}
$query_4="select * from tb_mr_bccd limit $page_4,1";
$result4=$conn->prepare($query_4);                    //准备查询语句
$result4->execute();                                  //执行查询语句，并返回结果集
$res4=$result4->fetch(PDO::FETCH_ASSOC);
echo $res4['tb_bccd_name'];
?></a></td>
```

PHP 上下分页中的其他内容已经在第 6.6.2 节中介绍过了，这里不再赘述，完整程序代码请参考本章光盘中的内容。

6.8 技 术 提 炼

6.8.1 自定义函数替换超长文本中的特殊字符技术

在输出超长文本中的数据时，应用自定义函数 unhtml()实现对超长文本中的特殊字符进行替换。unhtml()函数的语法如下：





```
function unhtml($content){                                          //定义自定义函数的名称
    $content=htmlspecialchars($content);                            //转换文本中的特殊字符
    $content=str_replace(chr(13),"<br>",$content);                  //替换文本中的换行符
    $content=str_replace(chr(32)," ",$content);                //替换文本中的 
    $content=str_replace("[_[","<",$content];                       //替换文本中的大于号
    $content=str_replace(")_)",">",$content);                       //替换文本中的小于号
    $content=str_replace("|_|"," ",$content);                       //替换文本中的空格
    return trim($content);                                          //删除文本中首尾的空格
}
```

首先，通过 file_get_contents()函数读取超长文本中的数据，返回一个字符串变量$counter，unhtml()函数就是对这个字符串进行操作，替换字符串中的特殊字符，然后再执行其他操作。关键代码如下：

```
<?php
include("function.php");                                            //调用指定的文件
if($page){
$counter=file_get_contents("file/file.txt");                        //读取指定文件的内容
$length=strlen(unhtml($counter));                                   //获取字符串长度
$page_count=ceil($length/1050);                                     //统计每页的字符数
$c=msubstr($counter,0,($page-1)*1050);                              //计算上页
$c1=msubstr($counter,0,$page*1050);                                 //计算当前页
echo substr($c1,strlen($c),strlen($c1)-strlen($c));                 //输出当前页的内容
}
?>
```

file_get_contents()函数将指定文件中的内容读入到一个字符串。如果操作系统支持内存映射技术，还会应用内存映射技术来增强其读入文件的性能。该函数的语法如下：

```
string file_get_contents(string filename [, int use_include_path [, resource context]])
```

file_get_contents()函数的参数说明如表 6.1 所示。

表 6.1　file_get_contents()函数的参数说明

参　　数	说　　明
filename	必要参数。用于指定某文件的路径及名称
use_include_path	可选参数。如果将该参数设置为 true，PHP 会尝试按照 include_path 标准包含路径中的每个指向去打开文件
context	可选参数。设置提高文件流性能的一些选项

6.8.2　自定义函数控制超长文本的输出技术

PHP 超长文本分页显示，控制超长文本中的数据输出应用的是自定义函数 msubstr()，该函数的语法如下：

```
function msubstr($str,$start,$len){      //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
$tmpstr="";
```

```
$strlen=$start+$len;                    //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
for($i=0;$i<$strlen;$i++){              //通过 for 循环语句循环读取字符串
if(ord(substr($str,$i,1))>0xa0){        //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
$tmpstr.=substr($str,$i,2);             //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
$i++;                                   //变量自加 1
}else{                                  //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
$tmpstr.=substr($str,$i,1);}
}
return $tmpstr;                         //输出字符串
}
```

首先，应用 include 调用指定的包含文件 function.php；然后，应用 file_get_contents()函数读取指定文本文件中的数据，并应用自定义函数 unhtml()替换字符串中的特殊字符；最后，应用 msubstr()函数控制数据的输出。关键代码如下：

```
<?php
include("function.php");
if($page){
$counter=file_get_contents("file/file.txt");                    //读取文本文件中的数据，返回一个字符串
$length=strlen(unhtml($counter));                               //计算字符串的长度，替换特殊字符
$page_count=ceil($length/1050);                                 //对字符串进行分页处理
$c=msubstr($counter,0,($page-1)*1050);                          //通过自定义函数获取上一页中的数据
$c1=msubstr($counter,0,$page*1050);                             //通过自定义函数获取当前页中的数据
echo substr($c1,strlen($c),strlen($c1)-strlen($c));             //获取当前页中要输出的数据
}
?>
```

6.8.3 连接数据库类的创建

要实现数据库中数据的分页显示，首先必须与数据库建立连接，这里通过一个自定义的类来实现数据库的连接，其中操作数据库使用 PDO 技术。该类存储于 06\01\system\system.class.inc.php 文件中，其代码如下：

```
class ConnDB{                                                   //数据库连接类
var $dbtype;
var $host;
var $user;
var $pwd;
var $dbname;
function ConnDB($dbtype,$host,$user,$pwd,$dbname){              //构造方法
$this->dbtype=$dbtype;
$this->host=$host;
$this->user=$user;
$this->pwd=$pwd;
$this->dbname=$dbname;
}
```



```
function GetConnId(){                                      //实现数据库的连接并返回连接对象
if($this->dbtype=="mysql" || $this->dbtype=="mssql"){            //判断数据库类型
$dsn="$this->dbtype:host=$this->host;dbname=$this->dbname";      //定义连接语句
}else{
$dsn="$this->dbtype:dbname=$this->dbname";
}
try {
$conn = new PDO($dsn, $this->user, $this->pwd);          //初始化一个 PDO 对象，创建数据库连接对象$conn
$conn->query("set names utf8");                          //设置编码格式
return $conn;                                            //返回连接对象
} catch (PDOException $e) {
die ("Error!: " . $e->getMessage() . "<br/>");
}
}
}
```

6.8.4　操作数据库类的创建

数据库连接成功后，接下来进行数据库的操作，这里同样对 PDO 操作数据库的方法进行了封装，在 AdminDB 类中，可以执行数据的添加、更新、删除和查询操作。类文件存储于 06\01\system\system.class.inc.php 中，其代码如下：

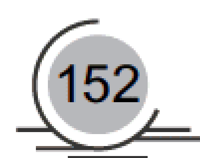
```
class AdminDB{                                           //数据库管理类
function ExecSQL($sqlstr,$conn){
$sqltype=strtolower(substr(trim($sqlstr),0,6));
$rs=$conn->prepare($sqlstr);                             //准备查询语句
$rs->execute();                                          //执行查询语句，并返回结果集
if($sqltype=="select"){
$array=$rs->fetchAll(PDO::FETCH_ASSOC);                  //获取结果集中的所有数据
if(count($array)==0 || $rs==false)
return false;
else
return $array;
}elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
if($rs)
return true;
else
return false;
}
}
}
```

6.8.5　分页类的创建

为了让分页技术的应用更加简单，可以将分页的方法写入一个类中，这样就可以在需要使用分页

的地方直接调用这个类实现分页的功能。分页类同样存储于 06\01\system\system.class.inc.php 文件中，其代码如下：

```
class SepPage{
var $rs;
var $pagesize;
var $nowpage;
var $array;
var $conn;
var $sqlstr;
function ShowData($sqlstr,$conn,$pagesize,$nowpage){                    //定义方法
if(!isset($nowpage) || $nowpage=="")                                    //判断变量值是否为空
$this->nowpage=1;                                                       //定义每页的起始页
else
$this->nowpage=$nowpage;
$this->pagesize=$pagesize;                                              //定义每页输出的记录数
$this->conn=$conn;                                                      //连接数据库返回的标识
$this->sqlstr=$sqlstr;                                                  //执行的查询语句
$offset=($this->nowpage-1)*$this->pagesize;
$sql=$this->sqlstr." limit $offset, $this->pagesize";
$result=$this->conn->prepare($sql);                                     //准备查询语句
$result->execute();                                                     //执行查询语句，并返回结果集
$this->array=$result->fetchAll(PDO::FETCH_ASSOC);                       //获取结果集中的所有数据
if(count($this->array)==0 || $this->array==false)
return false;
else
return $this->array;
}
function ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class){
$str="";
$res=$this->conn->prepare($this->sqlstr);                               //准备查询语句
$res->execute();                                                        //执行查询语句，并返回结果集
$this->array=$res->fetchAll(PDO::FETCH_ASSOC);                          //获取结果集中的所有数据
$record=count($this->array);                                            //统计记录总数
$pagecount=ceil($record/$this->pagesize);                               //计算共有几页
$str.=$contentname." ".$record." ".$utits." 每页 
".$this->pagesize." ".$utits." 第 ".$this->nowpage." 页/共 ".$pagecount." 页";
$str.="    ";
if($this->nowpage!=1)
$str.="<a href=".$_SERVER['PHP_SELF']."
?page=1&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">首页</a>";
else
$str.="<font color='#555555'>首页</font>";
$str.=" ";
if($this->nowpage!=1)
$str.="<a href=".$_SERVER['PHP_SELF']."?
page=".($this->nowpage-1)."
```



```
&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">上一页</a>";
else
$str.="<font color='#555555'>上一页</font>";
$str.=" ";
if($this->nowpage!=$pagecount)
$str.="<a href=".$_SERVER['PHP_SELF']."?page=".($this->nowpage+1)."
&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">下一页</a>";
else
$str.="<font color='#555555'>下一页</font>";
$str.=" ";
if($this->nowpage!=$pagecount)
$str.="<a href=".$_SERVER['PHP_SELF']."?page=".$pagecount."
&page_type=".$anothersearchstr."&parameter2=".$anothersearchstrs." class=".$class.">尾页</a>";
else
$str.="<font color='#555555'>尾页</font>";
if(count($this->array)==0 || $this->array==false)
return "无数据！";
else
return $str;
}
}
```

在这个分页类中，包含 ShowData($sqlstr,$conn,$pagesize,$nowpage)和 ShowPage($contentname,$utits,$anothersearchstr,$anothersearchstrs,$class)两个方法。其中 ShowData()方法根据参数传递的 SQL 语句，完成数据的分页查询操作，并且返回查询结果。在 ShowPage()方法中根据参数传递的值定义分页超链接的输出内容。

6.8.6　类的实例化

类创建成功之后，要想在程序中调用类中的方法完成指定的操作，必须先完成类的实例化操作，通过类实例化返回的对象才可以调用指定的方法。有关类的实例化操作在本模块中都统一存储于 06\01\system\system.inc.php 文件中，其代码如下：

```
<?php
require("system.smarty.inc.php");                //包含 Smarty 配置类
require("system.class.inc.php");                 //包含数据库连接和操作类
$connobj=new ConnDB("mysql","localhost","root","111","db_pagination"); //数据库连接类实例化
$conn=$connobj->GetConnId();                     //执行连接操作，返回连接标识
$admindb=new AdminDB();                          //数据库操作类实例化
$seppage=new SepPage();                          //分页类实例化
$usefun=new UseFun();                            //使用常用函数类实例化
$smarty=new SmartyProject();                     //调用 Smarty 模板
function unhtml($params){
extract($params);
$text=$content;
```

```
global $usefun;
return $usefun->UnHtml($text);
}
$smarty->register_function("unhtml","unhtml");        //注册模板函数
?>
```

6.9 本章小结

本章详细讲解了如何实现分页程序流程。同时，在分页显示的程序过程中，采用了 Ajax 无刷新分页技术，使页面没有全部跳转，只局部刷新。通过本章的学习，读者不仅可以了解 Ajax 技术的使用，而且可以熟悉分页显示的开发思想。

第 7 章

论坛模块

（PHP+MySQL 实现）

论坛已经成为时下一种比较普遍的网上交流手段，其表现形式多种多样，有个人论坛、企业论坛、图书论坛和软件论坛等，虽说名目众多，但都是围绕着一个“论”字在进行，其原理是相同的，都是对某个事物、事情或者人物发表自己的意见和看法。本章将详细讲解论坛的开发流程和在创建中使用的关键技术。

通过阅读本章，可以学习到：

- 小纸条信息无刷新输出
- 连接远程 MySQL 数据库
- 树状导航菜单
- 帖子的发布、回复
- 帖子置顶
- 帖子引用、收藏
- 屏蔽回帖

7.1 论坛模块概述

随着互联网的发展，网络信息资源也不断的丰富，而以动态性和交互性为特征的网络论坛是当中最丰富、最开放和最自由的网络信息资源，是最受欢迎的一种信息交流方式。目前实现论坛功能的开发语言有很多种，主要有 PHP、JSP、ASP 和 ASP.NET，其中 PHP 借助于开源的优势必将成为网络开发的必然趋势。

网络论坛和互联网上的其他信息一样，具有范围广、内容庞杂、动态变化性强等特点，其中两个最重要的特点是交互性和时效性。交互性是指论坛用户能够参加到论坛信息的交流过程中来，可以在论坛中发布自己的信息并且可以得到其他用户的反馈，这是网络论坛信息最基本和最重要的特征。一位用户发布信息，往往有很多人回应，可以开展多人讨论，这是一个多向交流信息的过程，它使人们能够便捷地进行交流。由于互联网信息传播的快捷与方便以及网络论坛的交互性，使得人们能够及时地从论坛中获取某一技术和行业的发展动态与最新的进展，专业技术论坛中信息的时效性更强，一些最新的技术信息都可以从论坛中获取。

7.1.1 系统功能结构

论坛是一个发布帖子和回复帖子的过程，根据论坛的基本功能，整理出论坛模块的功能结构图，如图 7.1 所示。

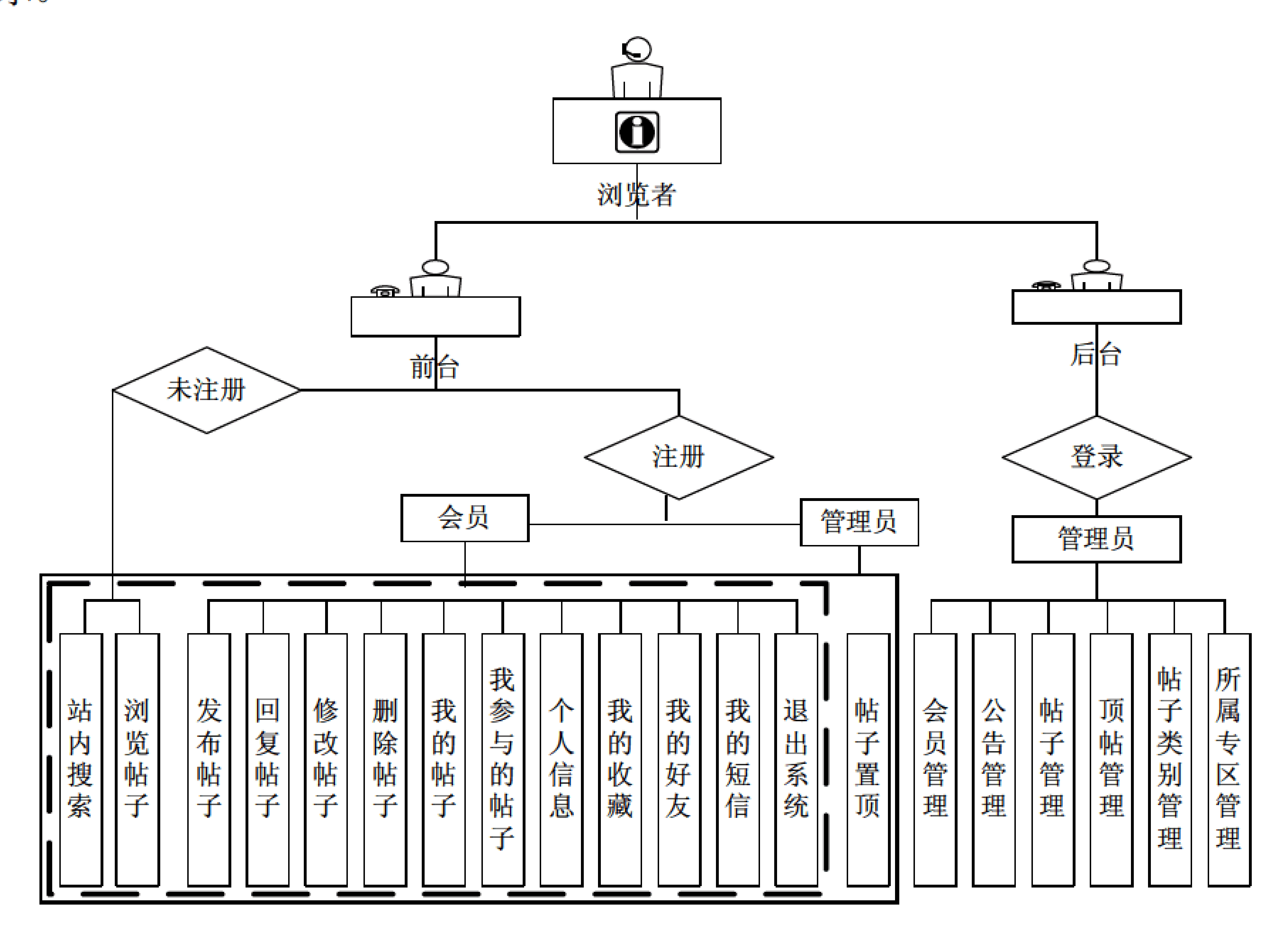
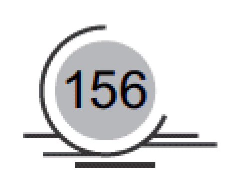


图 7.1 论坛模块的功能结构图

根据功能结构图中描述的功能，设计了一个完整的论坛模块的开发流程，如图 7.2 所示。



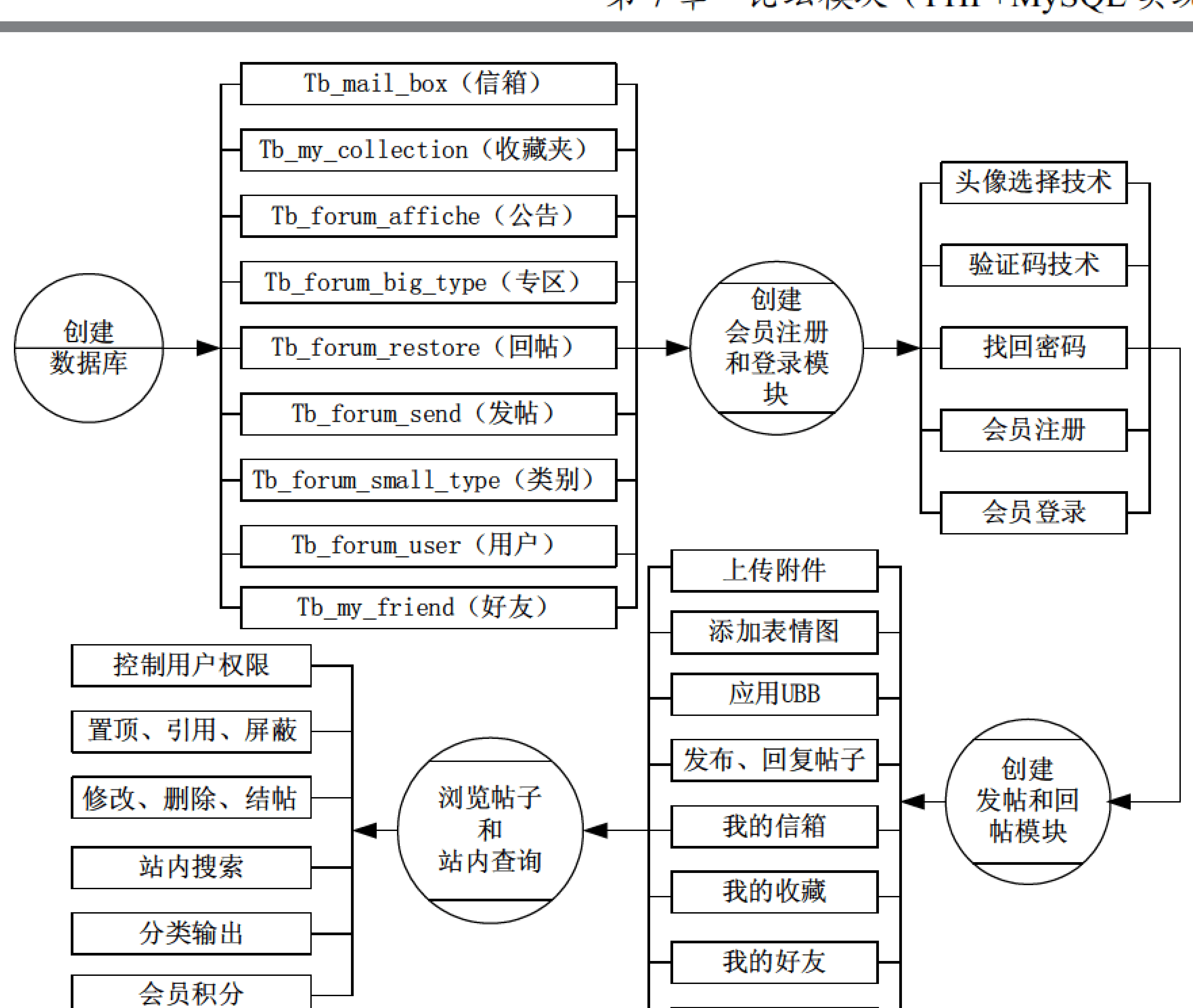


图 7.2　论坛模块的开发流程

7.1.2　程序预览

一个完整的论坛系统是由多个模块共同构建的，下面列出一些主要模块的运行结果，让读者对这个论坛模块有一个初步的了解和认识。

论坛首页运行效果如图 7.3 所示。其中包括树状导航菜单、用户登录、初始化信息等。



图 7.3　论坛首页运行效果

我的信箱运行结果如图 7.4 所示。其主要功能是收发站内邮件、接收系统信息、发送好友验证请求等操作，并可以在该模块中对个人收发的站内邮件进行管理。



图 7.4　我的信箱运行结果

用户登录论坛程序后，可以单击相关分类中的“我要提问”超链接发表新帖。在发表新帖的同时，该模块也支持个人附件的上传。发表新帖的运行结果如图 7.5 所示。

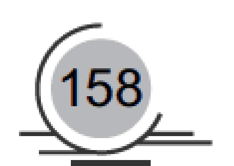


图 7.5　发表新帖的运行结果

在成功发表新帖后，新帖会被放置到该类帖子浏览页面的首位，这样其他用户可以在第一时间浏览到最新的帖子内容。其运行结果如图 7.6 所示。



图 7.6　最新帖子发布运行结果





为了使论坛功能更加合理、完美，本模块增加了帖子置顶、帖子引用、帖子收藏和屏蔽帖子等特殊功能，其中对于发表新帖的浏览和回复如图 7.7 所示。

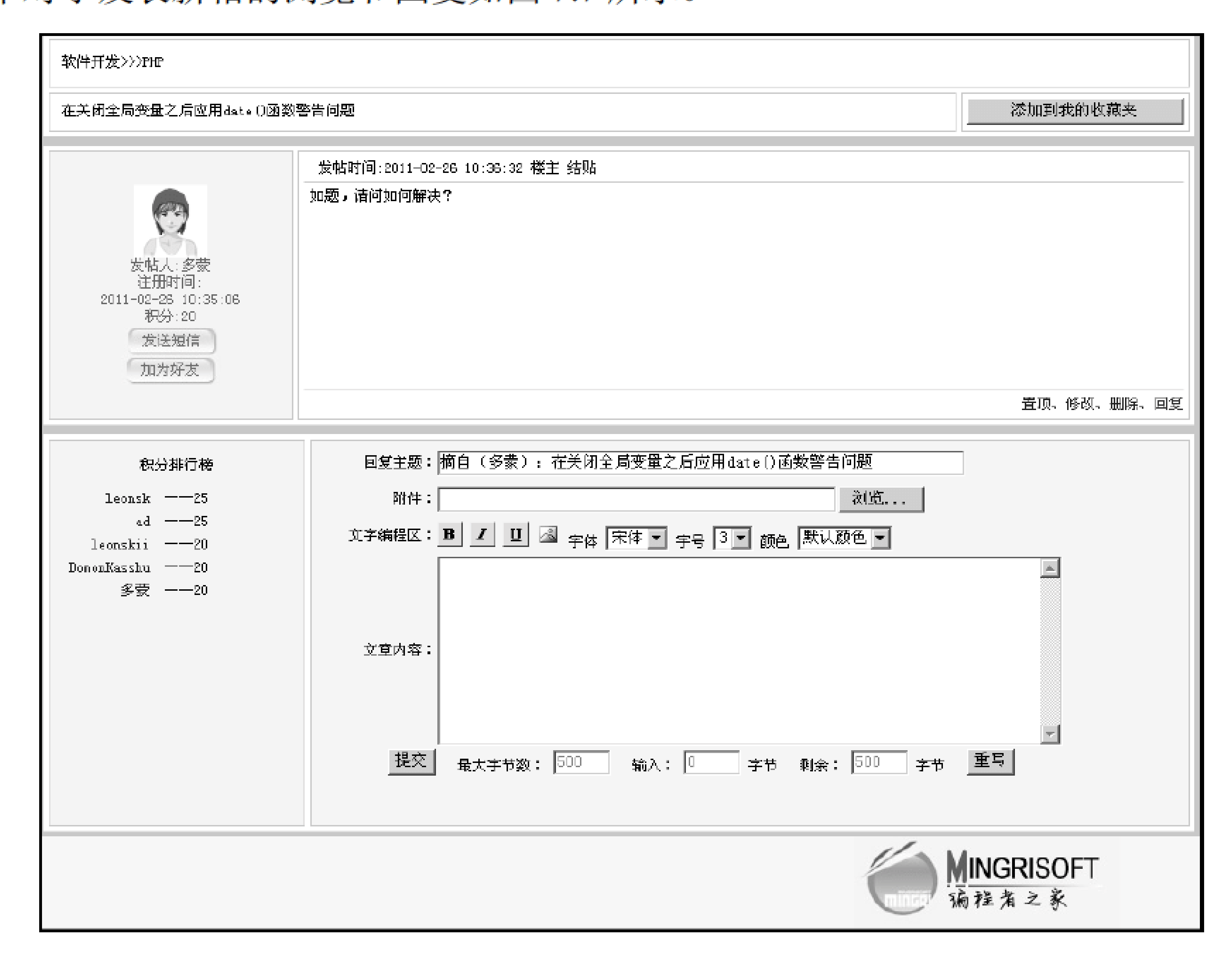


图 7.7　帖子的浏览和回复运行结果

论坛后台作为论坛的管理平台，在整个管理体系中占据着非常重要的位置，该模块主要用于管理用户的个人资料以及帖子的相关信息，并可以备份和恢复论坛系统的相关数据内容。论坛后台管理运行效果如图 7.8 所示。



图 7.8　论坛后台管理运行效果

另外，该论坛模块还包括一些辅助的功能，如我发布的帖子和我参与的帖子等。为了便于论坛管理，增加了管理员管理论坛的功能，包括发布帖子、回复帖子、帖子类别和置顶帖子以及数据的备份和恢复等内容，由于篇幅有限，这里不能一一列举。

7.2　热点关键技术

在论坛模块的开发过程中，有些关键的技术是不可缺少的，下面就对论坛中用到的关键技术进行详细介绍。

7.2.1　树状导航菜单

通过树状导航菜单能够对网站中的内容进行合理的分类处理，进而使网站的布局更加合理。这里将树状导航菜单的技术应用到论坛中帖子类别的处理上，改变了以往以整页篇幅输出帖子类别的设计思路，使论坛中帖子类别的输出更加合理、规范。其运行结果如图 7.9 所示。



图 7.9　通过树状菜单输出帖子类别

树状导航菜单的设计原理是：首先，从数据库中读取论坛中所属专区的数据进行输出；然后，通过 JavaScript 脚本语句控制单元格中内容的隐藏和显示；最后，根据所属的专区读取对应帖子类别的数据。其关键就是单元格属性的设置和 JavaScript 脚本语句的应用。

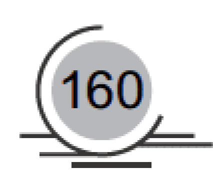
（1）在 left.php 文件中，首先从数据库的专区表中读取论坛中所属专区的数据。代码如下：

```
<?php    $query=mysql_query("select * from tb_forum_big_type");
         while($myrow=mysql_fetch_array($query)){      //循环输出数据库中的数据
              $querys=mysql_query("select * from tb_forum_small_type where tb_big_type_ content='$myrow
[tb_big_type_content]' ");                          //以所属专区为条件，读取对应专区中帖子的类别的数据
              $myrows=mysql_fetch_array($querys);
 ?>
```

（2）创建表格，输出所属专区的数据，应用 onClick 事件调用 JavaScript 语句 open_close()控制单元格中内容的隐藏和显示，其中还为单元格中的内容设置了超链接，链接到指定的文件。代码如下：

```
 <td width="84%" height="24" onClick="javascript:open_close(id_a<?php echo
$myrow[tb_big_type_id];?>);" >      <!--控制单元格的属性-->
    <!--设置超链接-->
    <a href="content.php?content=<?php echo $myrow["tb_big_type_content"];?>&&content_1=<?php echo
$myrows["tb_small_type_content"];?>" target="contentFrame"><?php echo $myrow["tb_big_type_content"];?>"
    </a>
    </td>
```

（3）创建表格，设置表格 ID 的值和 style 的值。以帖子的所属专区为查询条件，从类别数据表中读取符合条件的数据，将读取到的数据循环输出到单元格中。同样也为单元格中的内容设置了超链接，链接到指定的文件。代码如下：



```
<table width="170" border="0" cellpadding="0" cellspacing="0" bgcolor="#8394BF" id="id_a<?php echo $myrow
["tb_big_type_id"];?>" style="display:none"><!--设置表格的 ID 和样式-->
      <?php
            $query_1=mysql_query("select * from tb_forum_small_type where tb_big_type_content='$myrow[tb_
big_type_content]' ");
            while($myrow_1=mysql_fetch_array($query_1)){
      ?>
      <tr>
      <td align="right"> </td>
          <td height="23">  
                  <a href="content.php?content=<?php echo $myrow["tb_big_type_content"];?>&&content_1=<?php
echo $myrow_1["tb_small_type_content"];?>" target="contentFrame">
                        <?php echo $myrow_1["tb_small_type_content"];?></a></td>
      </tr>
      <?php }?></table>
<?php }?>
</td></tr></table>
```

其中，调用 JavaScript 脚本的代码如下：

```
<script language="javascript">
function open_close(x){                       //创建自定义函数
        if(x.style.display==""){              //判断当 x 的值为空时，将 x 的值更改为 none
        x.style.display="none";
      }else if(x.style.display=="none"){      //否则，当 x 的值等于 none 时，将 x 的值改为空
            x.style.display="";
      }
}
</script>
```

其主要功能是实现单元格的隐藏和显示，即当单元格处于显示状态时，单击单元格则执行隐藏的操作；当单元格处于隐藏状态时，单击单元格则执行显示的操作。

注意　在第 2 步的程序代码中，content.php 为指定链接的文件，其中栏目标识 content 为帖子的所属专区，而标识 content_1 则是帖子的类别。通过超链接的 target 属性指定链接文件的名称为 contentFrame。

至此，树状导航菜单技术介绍完毕。有关树状导航菜单的具体应用可以参考本章程序中的 left.php 文件。

7.2.2　帖子置顶

帖子置顶就是将指定的帖子在所有帖子的顶端显示，用于突出帖子的特殊性。只有管理员拥有帖子置顶的权限，其他任何人都不具备这个权限。帖子置顶的操作过程如图 7.10 所示。

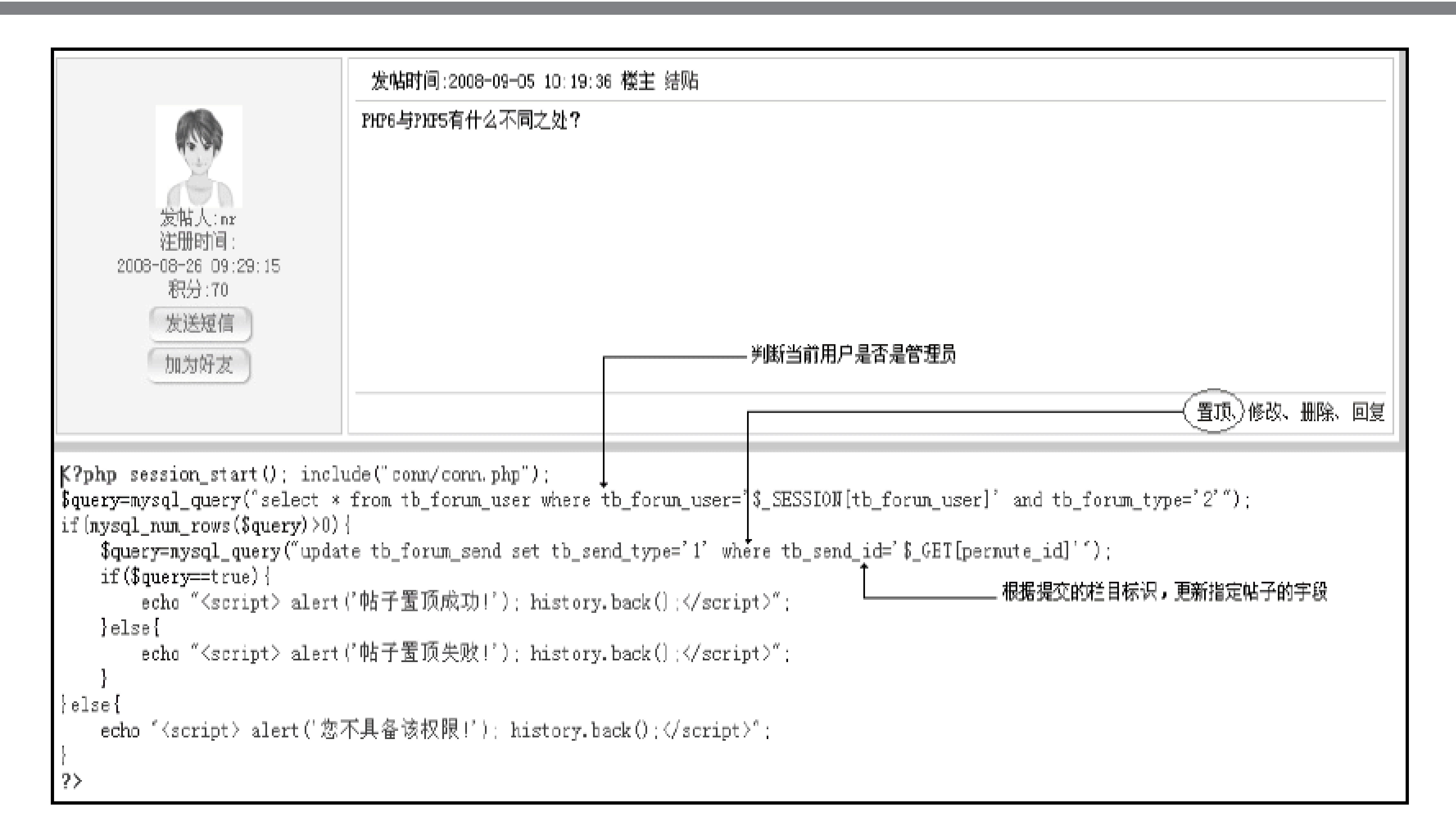


图 7.10 帖子置顶的操作过程

帖子置顶的操作原理是：根据帖子的 ID 设置一个超链接，在链接的 permute_send.php 文件中实现帖子置顶的操作。

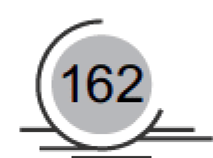
（1）在 send_forum_content.php 文件中，创建一个“置顶”超链接，链接的标识为对应帖子的 ID，链接到文件 permute_send.php 中，在该文件中实现帖子置顶的操作。关键代码如下：

```
<!--设置执行帖子置顶的超链接-->
<a href="permute_send.php?permute_id=<?php echo $myrow_3["tb_send_id"];?>">置顶</a>
```

（2）创建 permute_send.php 文件，实现帖子置顶的操作。首先，判断当前的用户是否是管理员，这里用户权限的设置是通过用户信息表（tb_forum_user）中 tb_forum_type 字段的值来控制的，如果 tb_forum_type 字段的值是 1，则代表注册的会员；如果值是 2，则代表是管理员。

如果当前用户是管理员，将指定帖子的 tb_send_type 字段的值更新为 1；否则不执行更新数据的操作，并且弹出提示信息“您不具备该权限!”，返回到上一页。相关代码如下：

```
<?php
session_start();
include("conn/conn.php");
$query=mysql_query("select * from tb_forum_user where tb_forum_user='$_SESSION[tb_forum_user]' and
tb_forum_type='2'");                              //判断当前用户是否是管理员
if(mysql_num_rows($query)>0){                     //当前用户是管理员，执行更新语句实现帖子置顶
    $query=mysql_query("update tb_forum_send set tb_send_type='1' where tb_send_id='$_GET[permute_id]'");
    if($query==true){
        echo "<script> alert('帖子置顶成功!'); history.back();</script>";
    }else{
        echo "<script> alert('帖子置顶失败!'); history.back();</script>";
    }
}else{
    echo "<script> alert('您不具备该权限!'); history.back();</script>";
}
?>
```



帖子置顶技术讲解完毕，有关该技术的完整应用可以参考后面小节中的程序。

7.2.3　帖子引用

帖子引用是指在浏览帖子时，针对某个回复的帖子或者自己的看法与楼上的看法相同，此时就可以单击引用超链接，直接将楼上回复的帖子进行引用，作为自己的回复帖子进行提交。帖子引用的操作流程如图 7.11 所示。

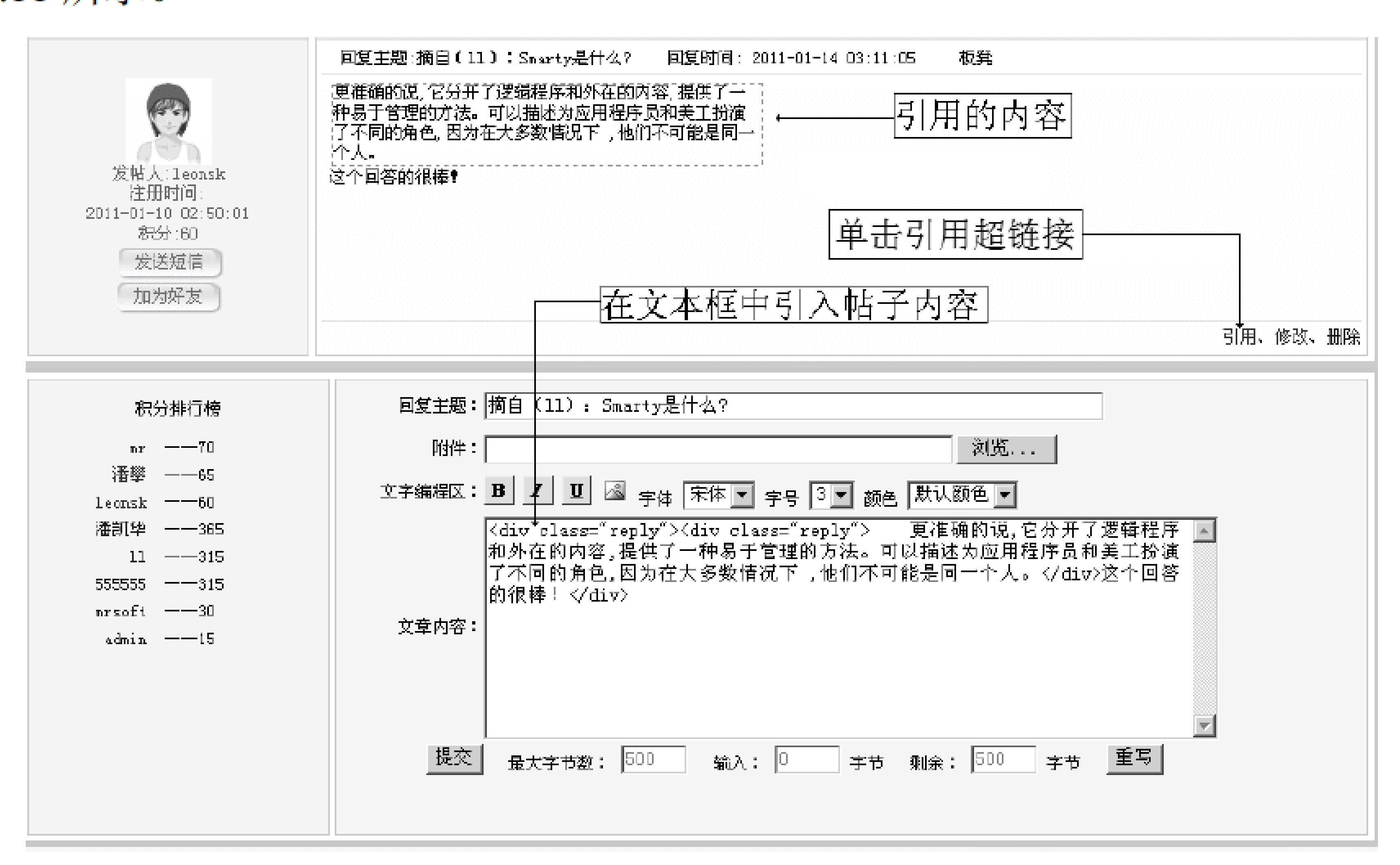


图 7.11　帖子引用的操作流程

帖子引用的实现原理是：首先，在帖子浏览的页面中针对每个回复的帖子设置一个“引用”超链接，这里将其链接到本页 send_forum_content.php 文件中，设置链接标识 cite 为回复帖子的 ID，添加锚点 bottom；然后，在指定输出回帖内容的表格中添加一个命名锚记，实现同一页面的引用跳转；最后，在输出引用内容的文本域中，根据超链接中传递的栏目标识，从数据库中读取到指定的回帖数据，将引用的内容进行输出。

（1）在 send_forum_content.php 文件中通过创建引用的超链接，并且设置链接的栏目标识 cite 和锚点 bottom 来实现引用的操作。关键代码如下：

```
<a href="send_forum_content.php?send_big_type=<?php echo
$_GET["send_big_type"];?>&&send_small_type=<?php echo $_GET["send_small_type"];?>&&send_id=<?php
echo $_GET["send_id"];?>&&cite=<?php echo $myrow_4["tb_restore_id"];?>#bottom">引用</a>
```

（2）在指定的位置设置一个命名锚记，实现同一页面的跳转。代码如下：

```
<a name="bottom" id="bottom"></a>
```

（3）在要输出引用内容的文本域中进行编辑，根据超链接栏目标识 cite 的值，从数据库中读取到对应的回复帖子的标题和内容，并且将读取的数据放入 div 标签后输出到文本域中，用于分隔引用和

回复的内容。代码如下：

```
<td width="641"><input name="restore_subject" type="text" id="restore_subject" value="
<?php
if($_GET["cite"]==true){
     $query=mysql_query("select * from tb_forum_send where tb_send_id='$_GET[send_id]'");
     $result=mysql_fetch_array($query);
echo "摘自（".$result["tb_send_user"]."）：".$result["tb_send_subject"];
}
?>" size="80">
</td>
<!--通过文本框输出引用内容-->
<td width="641"><textarea name="file" cols="80" rows="10" id="file" onKeyDown="countstrbyte (this.form.
file,this. form.total,this.form.used,this.form.remain);" onKeyUp=" countstrbyte (this.form. file,this.form.total, this.
orm.used, this.form.remain);">
<?php
    if($_GET["cite"]==true){
        $query=mysql_query("select * from tb_forum_restore where tb_restore_id='$_GET[cite]'");
        $result=mysql_fetch_array($query);      //根据 ID 从数据库中获取到引用的内容
        //输出引用内容放入类为 reply 的 div 内，用于分隔引用和回复的内容
        echo "<div class=\"reply\">".$result["tb_restore_content"]."</div>";
    }
    ?>
    </textarea>
    </td>
```

帖子引用技术是通过上述方法实现的，然后就可以将引用的内容直接进行提交，作为自己的回复帖子。有关该技术的完整应用可以参考本模块中的 send_forum_content.php 文件。

7.2.4 帖子收藏

帖子收藏就是将当前帖子的地址完整地保存到指定位置，为以后访问该帖子提供方便。帖子收藏的操作流程如图 7.12 和图 7.13 所示。

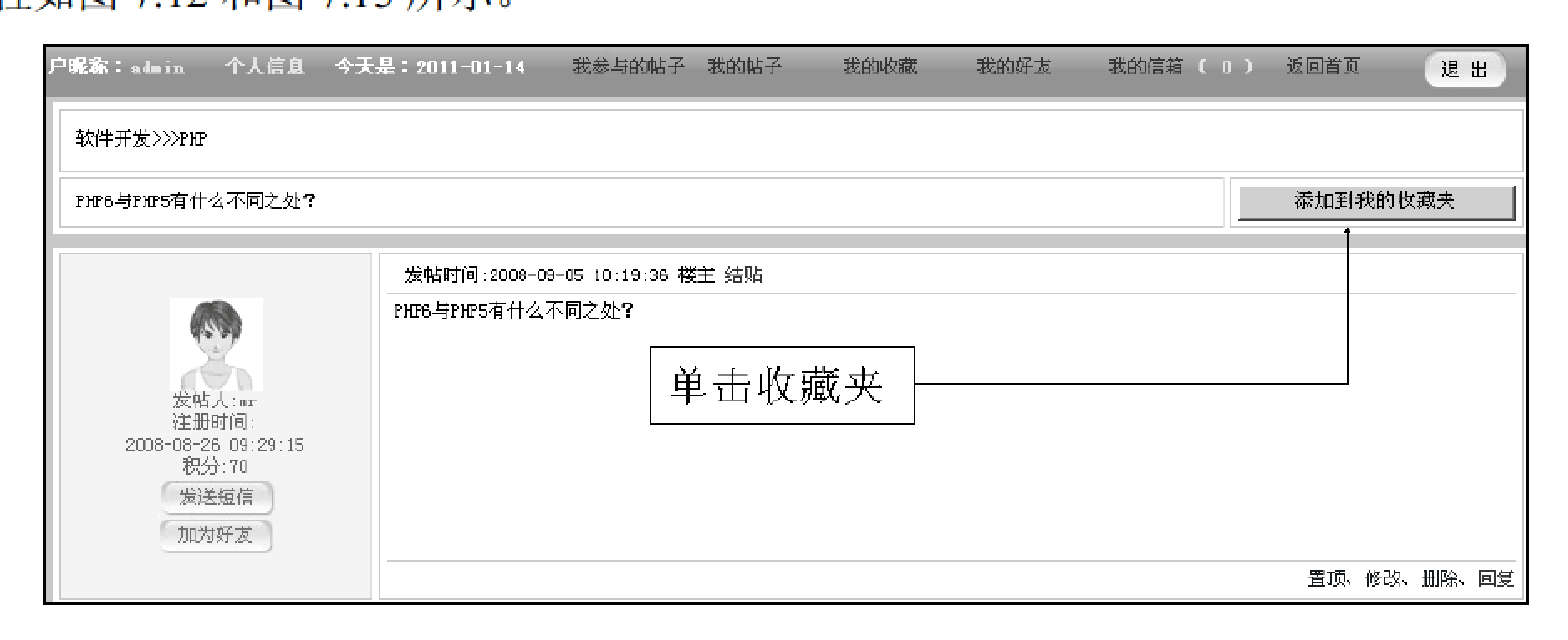


图 7.12　添加到我的收藏夹





图 7.13　获取帖子收藏地址

帖子收藏实现的关键是如何获取当前页面的完整地址。获取当前页面的完整地址主要是通过服务器变量$_SERVER 来获取收藏帖子的 URL 地址来实现的。

帖子收藏是在 send_forum_content.php 文件中实现的。关键代码如下：

```
$self=$_SERVER['HTTP_REFERER'];            //获取链接到当前页面的前一页面的 url 地址
$u=$_SERVER['HTTP_HOST'];                  //获取当前请求的 Host 头信息的内容
$r=$_SERVER['PHP_SELF'];                   //获取当前正在执行脚本的文件名
$l=$_SERVER['QUERY_STRING'];               //获取查询（query）的字符串（url 中第一个问号之后的内容）
$url="http://"."$u"."$r"."?"."$l";         //将获取的变量组成一个字符串，即完整的路径
```

在获取到完整的地址之后，接下来将帖子的标题、当前页面的完整路径和当前用户数据提交到 my_collection.php 页中，生成一个表单，最后将数据提交到 my_collection_ok.php 文件中，完成帖子的收藏。提交帖子完整地址和帖子标题的程序代码如下：

```
<?php if( !empty($_SESSION["tb_forum_user"])){   ?>
<form name="form1" method="post" action="my_collection.php?forum_subject=<?php echo
$myrow_3["tb_send_subject"];?>&&collection_user=<?php echo $_SESSION["tb_forum_user"];?>">
     <td width="173" height="22" align="center" valign="bottom">
     <input type="hidden" name="my_collection" value="<?php echo $url;?>">
     <input type="submit" name="Submit" value=" 添加到我的收藏夹 ">
</td>
</form>
<?php }   ?>
```

然后创建 my_collection.php 文件，生成一个表单，为收藏的帖子添加标签和说明，最后将数据提交到 my_collection_ok.php 文件中，将帖子收藏的数据添加到指定的数据表中。至此，帖子收藏技术讲解完毕，相关的程序代码请参考本书附带光盘中的内容，这里不再赘述。

7.2.5　屏蔽回帖

屏蔽回帖是管理员的权限，在论坛的后台管理中进行操作。帖子是否被屏蔽是根据回复帖子数据表中 tb_restore_tag 字段的值来判断的，如果帖子 tb_restore_tag 字段的值为 1，则说明该帖子被屏蔽；否则帖子没有被屏蔽。因此，屏蔽回帖就是将指定帖子的 tb_restore_tag 字段的值更新为 1。

屏蔽回帖主要通过两个文件来完成：一个是 message_restore.php，输出回复帖子的内容，创建执行

屏蔽帖子的 form 表单；另一个是 message_restore_ok.php，根据提交的数据，实现屏蔽帖子的操作。这里只给出屏蔽回帖的处理文件 message_restore_ok.php 的相关程序，其重要代码如下：

```
<?php session_start(); include("conn/conn.php");
$Submit=$_POST["Submit"];                                   //获取按钮值
if($Submit=="屏蔽"){                                        //判断当提交的值等于屏蔽时
    while(list($name,$value)=each($_POST)){                 //获取表单提交值并循环输出
        $result=mysql_query("update tb_forum_restore set tb_restore_tag='1' where tb_restore_id='".$name."'");
                                                            //根据 ID 的值执行循环操作
    if($result==true){
        echo "<script>alert('屏蔽成功!'); window.location.href='index.php?title=回帖管理';</script>";}}
}
if($Submit2=="取消"){                                       //判断当表单提交的值等于“取消”时
    while(list($name,$value)=each($_POST)){                 //获取表单提交的值
        $result=mysql_query("update tb_forum_restore set tb_restore_tag='0' where tb_restore_id='".$name."'");
                                                            //执行更新操作
    if($result==true){
        echo "<script>alert('取消屏蔽!'); window.location.href='index.php?title=回帖管理';</script>";}}
}
?>
```

7.2.6 连接远程 MySQL 数据库

在本模块中使用的是 MySQL 数据库，因此实现 PHP 与 MySQL 数据库的连接是一个非常关键的步骤，它是执行其他操作的先决条件。在开发程序的过程中，多数情况下使用的都是本地 MySQL 数据库，下面介绍一种连接远程 MySQL 数据库服务器的方法。

首先，在远程数据库服务器的机器中，通过 MySQL 数据库的图形化管理工具 phpMyAdmin，在权限中创建一个新用户，设置用户名为 root，密码为 111，主机为程序运行计算机的 IP 地址。操作步骤如图 7.14～图 7.16 所示。

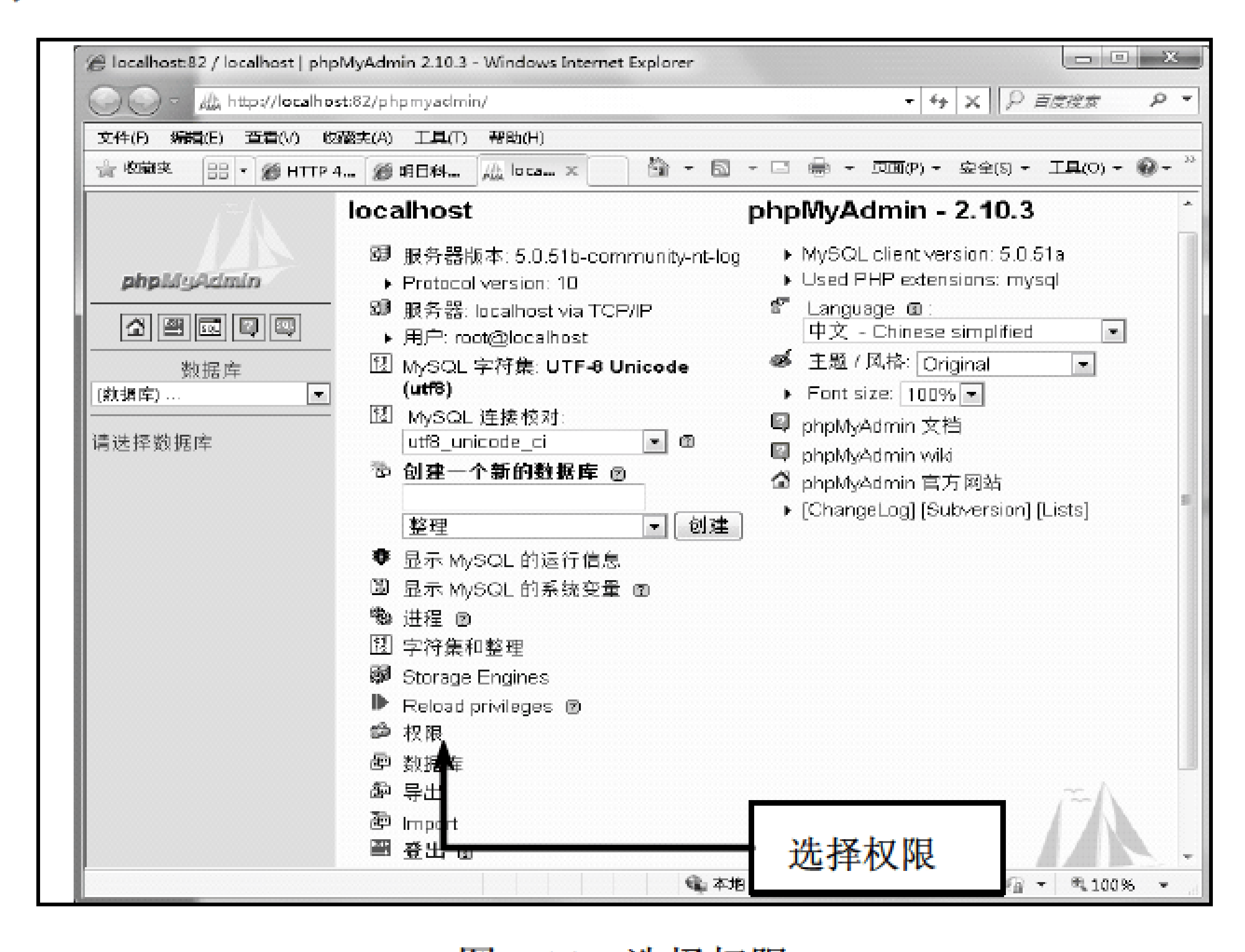
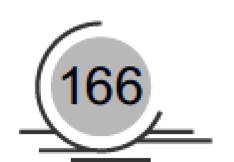


图 7.14　选择权限





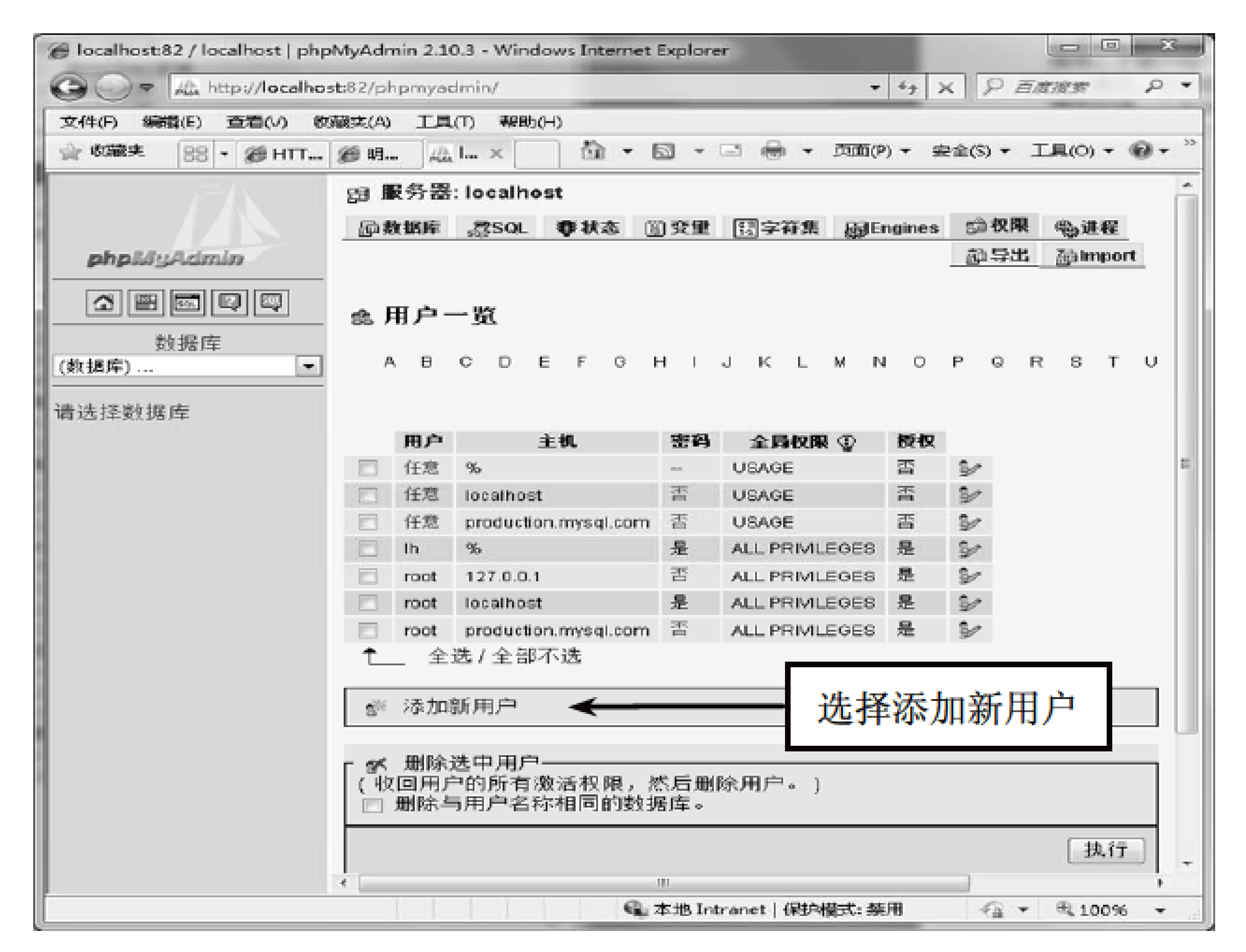


图 7.15　选择添加新用户

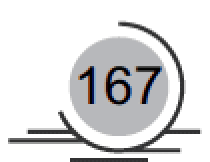


图 7.16　添加新用户

注意　远程 MySQL 数据库服务器所在计算机 IP 地址是 192.168.1.99。这里添加新用户的用户名是 root，密码是 111，主机是 192.168.1.149，192.168.1.149 是程序运行所在计算机的 IP 地址。

新用户创建完成之后连接远程的 MySQL 数据库服务器。使用的函数与连接本地服务器相同。程序代码如下：

```
<?php
    $conn=mysql_connect('192.168.1.99','root','111');          //连接数据库服务器 192.168.1.99
```

```
    mysql_select_db('db_forum',$conn);                    //连接 db_forum 数据库
    mysql_query("set names gb2312");                      //指定数据库中字符的编码格式
?>
```

注意 这里指定的服务器是远程 MySQL 数据库服务器的 IP 地址，用户名是 root，密码是 111。

7.2.7 小纸条信息的无刷新输出

小纸条信息的无刷新输出主要应用的是 Ajax 技术。通过 Ajax 技术调用指定的文件查询是否存在新的消息，并且将结果返回，通过 div 输出 Ajax 中返回的查询结果。运行结果如图 7.17 所示。

我参与的帖子　我的帖子　我的收藏　我的好友　我的信箱 （ 0 ）　返回首页

图 7.17　小纸条信息无刷新输出

关键的 Ajax 代码如下：

```
<script type="text/javascript" src="js/xmlHttpRequest.js"></script>
<script language="javascript">
function show_counts(sender){                  //通过 show_counts()函数调用指定的文件统计最新信息
url='show_counts.php?sender='+sender;          //设置要执行的文件，并设置栏目标识
xmlHttp.open("get",url, true);                 //执行操作
    xmlHttp.onreadystatechange = function(){
        if(xmlHttp.readyState == 4){
            tet = xmlHttp.responseText;
            show_counts11.innerhtml=tet;       //返回统计结果
        }
}
xmlHttp.send(null);
}
</script>
<!--间隔 1000 毫秒执行一次 show_counts()函数-->
<script language="javascript">
setInterval("show_counts('<?php echo $_SESSION["tb_forum_user"];?>')",1000);
</script>
<!--输出最新消息使用的是 div 标签-->
<td width="15" align="center" class="STYLE2"><div id="show_counts11"></div></td>
```

7.2.8 清除个人站内邮件

一个完善的论坛系统离不开站内邮件的应用，会员在浏览信件后，可能产生大量过期邮件，这些过期数据既给用户在邮件查找上带来麻烦，又给数据库增添了额外负担。这时用户可以通过筛选，对站内邮件有选择地删除。站内邮件的删除过程如图 7.18 所示。



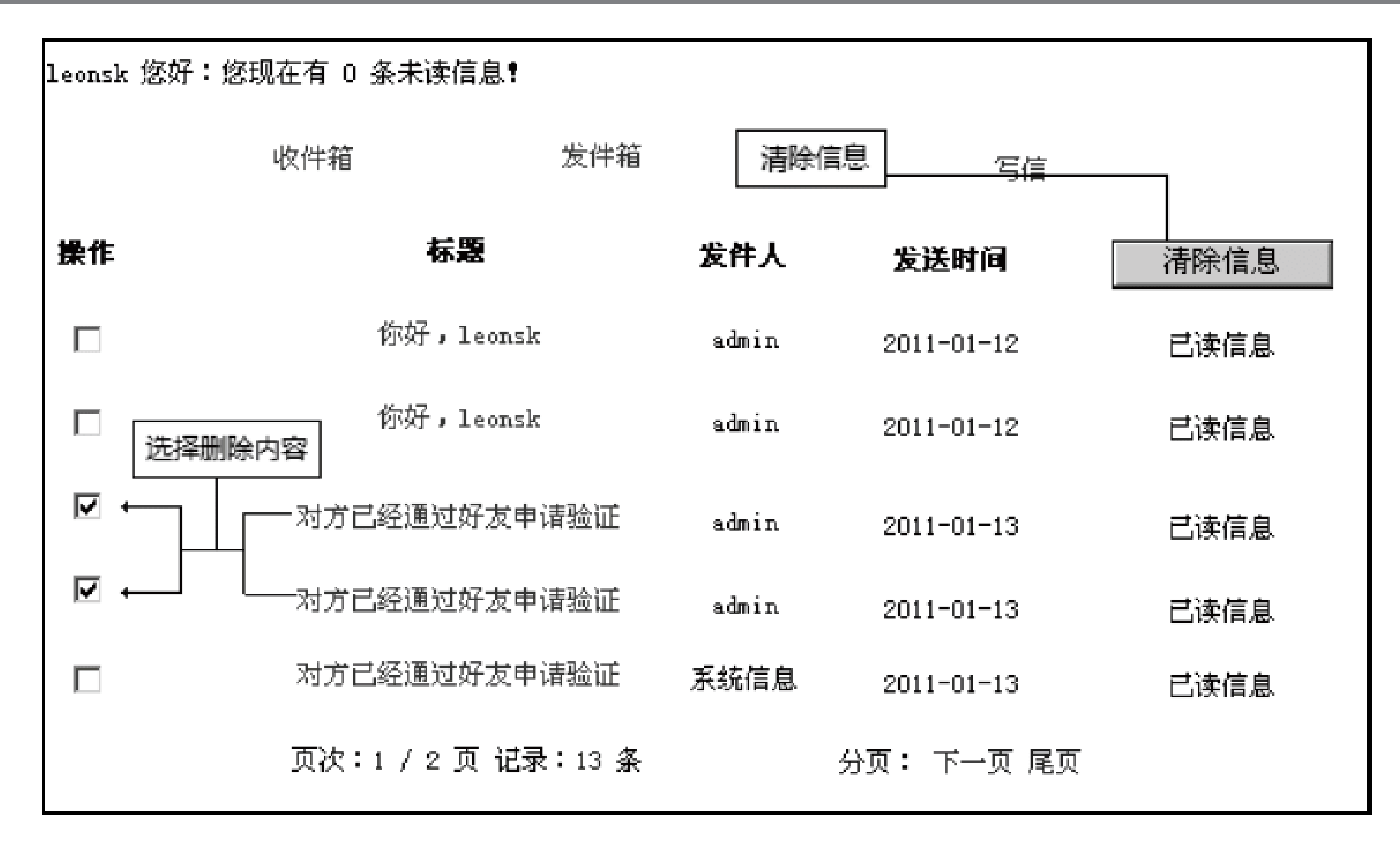


图 7.18　站内邮件的删除过程

获取要删除的站内邮件 id 后，将选定的 id 值提交到 delete_mail.php 文件中，完成个人站内邮件删除操作。选择删除站内邮件的文件 browse_mail.php 的关键代码如下：

```
<form name="form1" method="post" action="delete_mail.php?sender=<?php echo $_GET["sender"];?>&&mails=
<?php echo $_GET["mails"];?>">
  <tr>
    <td width="77" height="35" align="center"><strong><span class="STYLE_mail">操作</span></strong></td>
    <td width="266" align="center"><strong><span class="STYLE_mail">标题</span></strong></td>
    <td width="178" align="center"><strong><span class="STYLE_mail">发件人</span></strong></td>
    <td width="206" align="center"><strong><span class="STYLE_mail">发送时间</span></strong></td>
    <td width="189" height="35" align="center" class="STYLE_mail"><input name="Submit" type="submit"
id="Submit" value=" 清除信息 "></td>
  </tr>
<?php
  if($page){
    $page_size=10;                                    //每页显示 10 条记录
    //从数据库中读取数据
    $query="select count(*) as total from tb_mail_box where tb_receiving_person='$_GET[sender]' ";
$result=mysql_query($query);
    $message_count=mysql_result($result,0,"total");      //获取总的记录数
    $page_count=ceil($message_count/$page_size);         //获取总的页数
    $offset=($page-1)*$page_size;
$query=mysql_query("select * from tb_mail_box where tb_receiving_person='$_GET[sender]' order by tb_mail_
type='1' limit $offset, $page_size");
while($myrow=mysql_fetch_array($query)){
?>
  <tr>
    <td height="35" align="center"><input name="<?php echo $myrow["tb_mail_id"];?>" type="checkbox" value=
"<?php echo $myrow["tb_mail_id"];?>">    </td>
    <td align="center"><a href="browse_mail_content.php?mail_id=<?php echo $myrow["tb_mail_id"];?>" target=
"_blank" class="STYLE3_mail"><?php echo $myrow["tb_mail_subject"];?></a></td>
```

```
        <td align="center"><span class="STYLE3_mail"><?php echo $myrow["tb_mail_sender"];?></span></td>
        <td align="center"><span class="STYLE3_mail"><?php echo $myrow["tb_mail_date"];?></span></td>
        <td align="center"><span class="STYLE3_mail">
          <?php if($myrow["tb_mail_type"]=='1'){echo "已读信息";}else{echo "未读信息";}?>
        </span></td>
    </tr>
<?php }}?>
</form>
```

在获取要删除邮件的 id 后，将 id 提交到 delete_mail.php 页中，完成选中邮件的删除操作。删除站内邮件的程序代码如下：

```
<?php session_start(); include("conn/conn.php");
$Submit=$_POST['Submit'];                          //获取提交的 Submit 值
$mails=$_GET['mails'];
if($Submit==" 清除信息 "){
      while(list($name,$value)=each($_POST)){
           $array[]=$name;                          //将获得的 id 元素添加到数组中
            }
            if(count($array)-1>0){                  //如果获得的总数减 1 大于 0，则说明有选中的删除元素
                 for($i=1;$i<count($array);$i++){   //循环删除选中的信息
                      $result=mysql_query("delete from tb_mail_box where tb_mail_id='".$array[$i]."'");
                      }
              echo "<script>alert('删除成功!'); window.location.href='send_mail.php?sender=$_GET[sender]
&&mails=$mails';</script>";
           }else{
               echo "<script>alert('请先选择要删除的信件!');history.back();</script>";    //如果没有选择删除的
文件，则给出一个提示
            }
}
?>
```

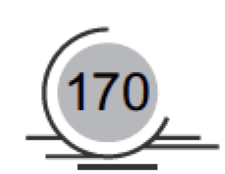
首先获取从 browse_mail.php 中得到的 Submit 传递值和选中邮件的 id 值，如果该页面获得的 Submit 传递值为“清除信息”，则继续判断选中邮件 id 的数量。

说明 利用 POST 方式传递相关参数，其中前台表单域的 Submit 也是利用该方式进行传递，所以 Submit 也会在后台处理页作为一个元素被添加到数组中，在判断选定 id 值的个数时，需要执行减 1 操作，在执行循环操作的过程中也需要跳过此数组中的值，以免得到不必要的结果。

然后，判断选择邮件 id 的数量，如果选择邮件 id 的数量为 0，则给出当前会员警告提示，告之其不能进行删除操作；如果判断选择的邮件 id 数量不为 0，则继续向下执行其他操作。

最后，通过选择的 id 来删除数据库中的邮件内容，从而实现邮件删除操作。

至此，站内邮件清除技术讲解完毕，此技术同样应用于好友删除技术中。





7.3　数据库设计

7.3.1　数据库分析

论坛的功能完善与否，数据库的运用是一个决定性的因素。只有拥有一个强大数据库的支持，论坛的功能才能够展现，否则它将和留言簿没什么区别。

本论坛中使用的是一个名称为 db_forum 的数据库，在该数据库中有 9 个数据表。有关数据表名称及表功能的介绍如图 7.19 所示。



服务器: localhost ▸ 数据库: db_forum

结构 SQL 搜索 查询 导出 Import 操作 权限 删除

表	类型	整理	说明
tb_forum_affiche	MyISAM	gb2312_chinese_ci	存储公告信息的数据表
tb_forum_big_type	MyISAM	gb2312_chinese_ci	存储所属专区的类别
tb_forum_restore	MyISAM	gb2312_chinese_ci	存储论坛回复的数据
tb_forum_send	MyISAM	gb2312_chinese_ci	存储论坛发送的数据
tb_forum_small_type	MyISAM	gb2312_chinese_ci	存储论坛中帖子的类别
tb_forum_user	MyISAM	gb2312_chinese_ci	存储用户的注册信息
tb_mail_box	MyISAM	gb2312_chinese_ci	存储短信信息
tb_my_collection	MyISAM	gb2312_chinese_ci	存储我收藏的数据
tb_my_friend	MyISAM	gb2312_chinese_ci	存储我的好友信息

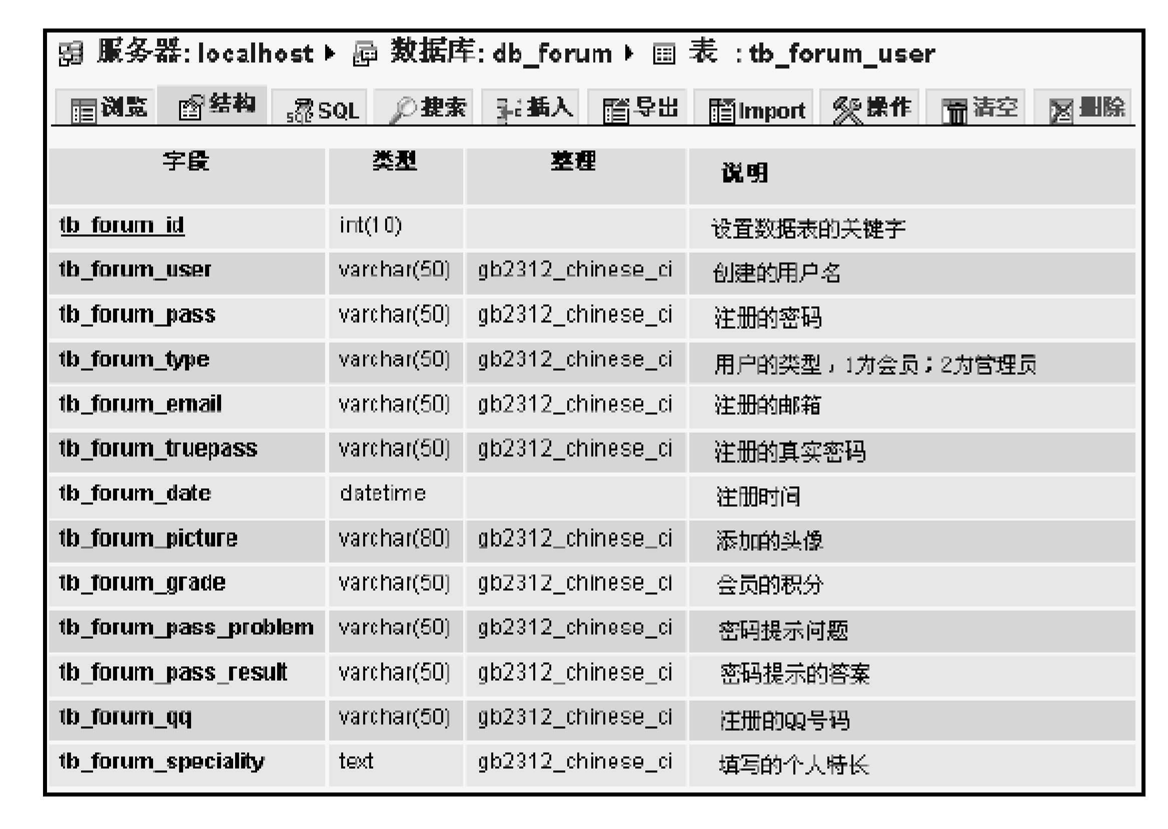
图 7.19　db_forum 数据库中的数据表

7.3.2　创建数据库中的数据表

下面对数据库中几个相对比较复杂的数据表的功能和结构进行介绍。

☑　tb_forum_user 数据表：用于存储用户的注册信息。其中包括 13 个字段，字段属性的说明如图 7.20 所示。

☑　tb_forum_send 数据表：用于存储论坛中发布帖子的数据。其中包括 11 个字段，字段属性的说明如图 7.21 所示。



服务器: localhost ▸ 数据库: db_forum ▸ 表 : tb_forum_user

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	说明
tb_forum_id	int(10)		设置数据表的关键字
tb_forum_user	varchar(50)	gb2312_chinese_ci	创建的用户名
tb_forum_pass	varchar(50)	gb2312_chinese_ci	注册的密码
tb_forum_type	varchar(50)	gb2312_chinese_ci	用户的类型，1为会员；2为管理员
tb_forum_email	varchar(50)	gb2312_chinese_ci	注册的邮箱
tb_forum_truepass	varchar(50)	gb2312_chinese_ci	注册的真实密码
tb_forum_date	datetime		注册时间
tb_forum_picture	varchar(80)	gb2312_chinese_ci	添加的头像
tb_forum_grade	varchar(50)	gb2312_chinese_ci	会员的积分
tb_forum_pass_problem	varchar(50)	gb2312_chinese_ci	密码提示问题
tb_forum_pass_result	varchar(50)	gb2312_chinese_ci	密码提示的答案
tb_forum_qq	varchar(50)	gb2312_chinese_ci	注册的QQ号码
tb_forum_speciality	text	gb2312_chinese_ci	填写的个人特长

图 7.20　tb_forum_user 数据表



服务器: localhost ▸ 数据库: db_forum ▸ 表 : tb_forum_send

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	说明
tb_send_id	int(10)		设置数据表的主键
tb_send_subject	varchar(50)	gb2312_chinese_ci	发布帖子的主题
tb_send_content	mediumtext	gb2312_chinese_ci	发布帖子的内容
tb_send_user	varchar(50)	gb2312_chinese_ci	帖子的发布者
tb_send_date	datetime		帖子的发布时间
tb_send_picture	varchar(70)	gb2312_chinese_ci	表情图
tb_send_type	varchar(50)	gb2312_chinese_ci	判断帖子是否置顶，1为置顶
tb_send_types	varchar(50)	gb2312_chinese_ci	判断帖子是否已经回复，1为回复
tb_send_small_type	varchar(50)	gb2312_chinese_ci	帖子所属的类别
tb_send_type_distillate	varchar(50)	gb2312_chinese_ci	判断帖子是否为精华帖，1为精华帖
tb_send_type_hotspot	varchar(50)	gb2312_chinese_ci	判断帖子是否为热门帖，1为热门帖

图 7.21　tb_forum_send 数据表

☑ tb_forum_restore 数据表：用于存储论坛中回复帖子的数据。其中包括 7 个字段，字段属性的说明如图 7.22 所示。

服务器: localhost ▸ 数据库: db_forum ▸ 表 : tb_forum_restore

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	说明
tb_restore_id	int(10)		设置数据表的主键
tb_restore_subject	varchar(50)	gb2312_chinese_ci	回复帖子的主题
tb_restore_content	mediumtext	gb2312_chinese_ci	回复帖子的内容
tb_restore_user	varchar(50)	gb2312_chinese_ci	回复者
tb_send_id	int(10)		发布帖子的ID
tb_restore_date	datetime		回复时间
tb_forum_counts	varchar(50)	gb2312_chinese_ci	帖子回复的次数

图 7.22　tb_forum_restore 数据表

☑ tb_my_collection 数据表：用于存储用户收藏的帖子。其中包括 7 个字段，字段属性的说明如图 7.23 所示。

服务器: localhost ▸ 数据库: db_forum ▸ 表 : tb_my_collection

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	说明
tb_collection_id	int(10)		设置数据表的ID
tb_collection_subject	varchar(50)	gb2312_chinese_ci	收藏帖子的标题
tb_collection_address	varchar(150)	gb2312_chinese_ci	收藏帖子的地址
tb_collection_label	varchar(50)	gb2312_chinese_ci	收藏帖子的标签
tb_collection_summary	mediumtext	gb2312_chinese_ci	收藏帖子的说明
tb_collection_user	varchar(50)	gb2312_chinese_ci	收藏者
tb_collection_date	datetime		收藏时间

图 7.23　tb_my_collection 数据表

在本模块中包括 9 个数据表，由于篇幅所限，这里只介绍了其中 4 个相对比较复杂的数据表，有关其他数据表的内容和属性可以参考本书附带光盘。

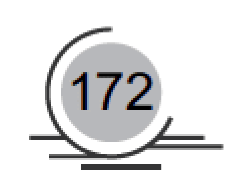
7.4　帖子的发布、浏览和回复

7.4.1　帖子的发布、浏览和回复概述

本论坛模块帖子功能包括帖子的发布、帖子的浏览以及帖子的回复等。其中，帖子发布为登录的会员提供一个发布帖子的操作平台。在该平台中，登录用户可以选择发布帖子的类别、自定义帖子的主题、选择表情图、选择上传附件以及通过文本编辑器对发布帖子的内容进行编辑。帖子发布的运行结果如图 7.24 所示。

帖子回复实现对指定的帖子进行回复的操作，帖子回复的运行结果如图 7.25 所示。

帖子浏览包括帖子主题和帖子内容的浏览。首先可以浏览到根据不同类别进行划分的帖子主题，然后可以在相应的帖子主题中浏览到具体帖子的内容。帖子主题和帖子内容浏览的运行结果如图 7.26





和图 7.27 所示。



图 7.24　帖子发布的运行结果

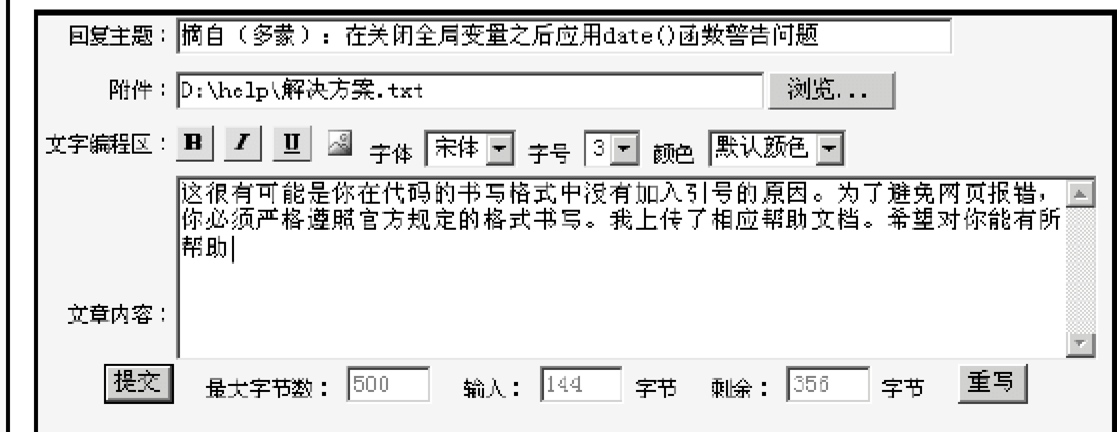


图 7.25　帖子回复的运行结果



图 7.26　帖子主题浏览的运行结果

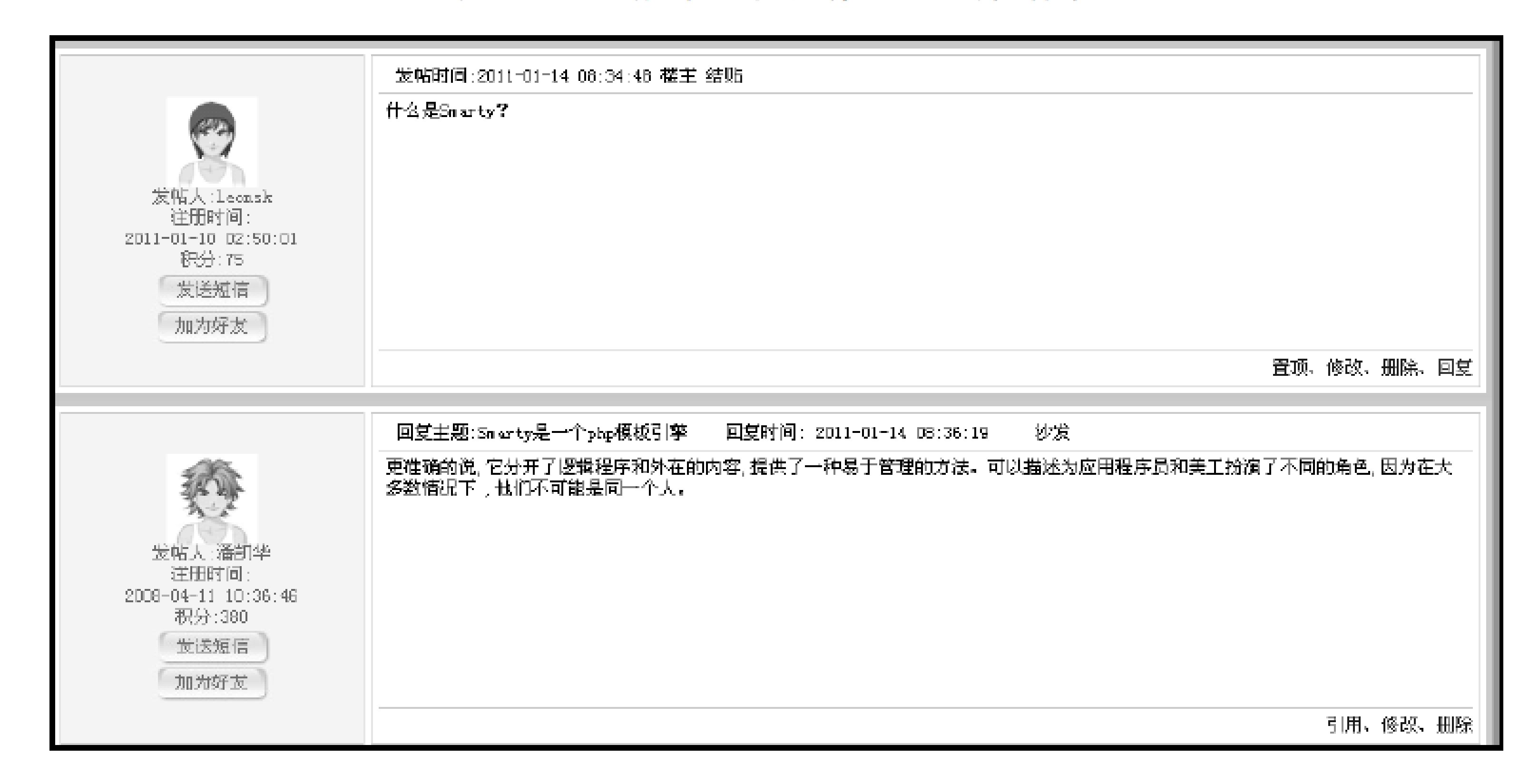


图 7.27　帖子内容浏览的运行结果

7.4.2 帖子发布功能实现

帖子发布主要由两个文件构成：一个是帖子发布内容的填写文件 send_forum.php；另一个是提交数据的处理文件 send_forum_ok.php。

在 send_forum.php 文件中，可以将该文件中的内容分成 3 个部分：第 1 部分初始化 Session 变量，连接数据库以及调用 js 文件；第 2 部分输出当前登录会员的个人信息；第 3 部分构建 from 表单，实现发布帖子数据的提交。

（1）初始化 Session 变量，连接数据库以及调用指定的包含文件，并且判断当前用户是否是会员，如果不是会员将不能进行帖子发布的操作。

帖子的发布文件 send_forum.php 的关键代码如下：

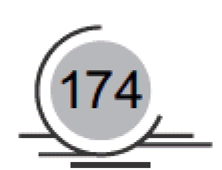
```
<?php session_start(); include("conn/conn.php");
if($_SESSION["tb_forum_user"]==true){          //判断当以会员登录时执行下面的内容
?>
<script type="text/javascript" src="js/editor.js"></script>
```

（2）从数据库中读取出当前会员的个人信息，并且将个人信息输出。代码如下：

```
<?php $query_1=mysql_query("select * from tb_forum_user where tb_forum_user='$_SESSION["tb_forum_
user"]'",$conn);
$myrow_1=mysql_fetch_array($query_1); //执行查询语句，获取当前登录用户的个人信息
echo "<img src='$myrow_1["tb_forum_picture"]'>";
echo "<br>";
echo "当前用户:";
echo "<br>";
echo $myrow_1 ["tb_forum_user"];
echo "<br>";
echo "注册时间:";
echo "<br>";
echo $myrow_1["tb_forum_date"];
echo "<br>";
echo "积分:";
echo $myrow_1["tb_forum_grade"];
?>
```

（3）创建 form 表单，提交发布帖子的数据，包括帖子的类别、主题、表情图、文本内容和发布人。在对帖子内容进行填写时，应用的是一个文本编辑器，通过文本编辑器可以对提交的内容进行编辑。程序关键代码如下：

```
<!--将帖子类别输出到下拉列表框中-->
<td align="center">帖子类别:</td>
<td><select name="send_sort" id="send_sort">
     <option selected="selected">请选择帖子的类别</option>
     <?php      //输出帖子的类别
          $query_2=mysql_query("select * from tb_forum_small_type");
```



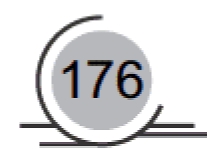
```
            while($myrow_2=mysql_fetch_array($query_2)){
    ?>
                <option value="<?php echo $myrow_2["tb_small_type_content"];?>">
                    <?php echo $myrow_2["tb_small_type_content"];?>
                </option>
        <?php }?>
    </select>
</td>
<!--表情图的排列输出-->
<td align="right" class="STYLE11">表情图：</td>
    <td><table>
    <tr>
        <td height="80" colspan="2"><div align="center">
            <table height="30" border="0" align="center" cellpadding="0" cellspacing="0">
                <tr>
                    <?php
                    for($i=1;$i<=24;$i++){          //根据文件夹中表情图的个数创建循环语句
                        if($i%6==0){                //判断变量的值是否等于 0
                    ?>
                    <td width="40" height="30"><div align="center"><!--输出表情图-->
                        <img src=<?php echo("images/inchoative/face".($i-1).".gif");?> width="20" height=
"20"></div></td>
                    <td width="40" height="30"><div align="center"><!--创建单选按钮-->
                        <input type="radio" name="face" value="<?php echo " ("images/ inchoative/
face.($i-1).".gif");?>">
                    </div></td>
                </tr>
                    <?php }else{?>
                    <td width="40" height="30"><div align="center">
                        <img src=<?php echo("images/inchoative/face".($i-1).".gif");?> width="20" height=
"20"></div></td>
                    <td width="40" height="30"><div align="center">
                        <input type="radio" name="face" value="<?php echo("images/ inchoative/face".
($i-1).".gif");?>" <?php if($i==1) { echo "checked";}?>>
                    </div></td>
                    <?php   }  }?>
            </table>
          </div></td>
    </tr>
</table></td>
<!--通过文本编辑器，对帖子的内容进行编辑-->
<tr>
    <td width="107" align="right" class="STYLE11">文章内容：</td>
    <td width="569">
<textarea name="menu" cols="1" rows="1" id="menu" style="position:absolute;left:0;visibility:hidden;"></textarea>
<script type="text/javascript">
var editor = new FtEditor("editor");
editor.hiddenName = "menu";
editor.editorWidth = "100%";
```

```
editor.editorHeight = "300px";
editor.show();
</script>
<input type="hidden" name="tb_forum_user" value="<?php echo $_SESSION["tb_forum_user"];?>"></td>
</tr>
```

有关 send_forum.php 文件的讲解到此结束，完整代码请参考本书附带光盘中的内容。

下面介绍表单处理页 send_forum_ok.php 文件。在该文件中将表单中提交的数据存储到数据库中，完成发布帖子信息的存储。帖子发布处理页 send_forum_ok.php 文件的关键代码如下：

```
<?php session_start(); include_once("conn/conn.php");
$tb_send_type=0;                                         //设置帖子是否置顶
$tb_send_types=0;                                        //判断帖子是否有回复
$tb_send_small_type=$_POST["send_sort"];                 //获取表单中提交的数据
$tb_send_subject=$_POST["send_subject"];                 //获取表单中提交的数据
$tb_send_picture=$_POST["face"];                         //获取表单中提交的数据
$tb_send_content=trim($_POST["menu"]);                   //获取表单中提交的数据
$tb_send_user=$_POST["tb_forum_user"];
$tb_send_date=date("Y-m-j H:i:s");
if($_FILES[send_accessories][size]==0){                  //判断是否有附件上传
    //执行添加语句
   if(mysql_query("insert into tb_forum_send(tb_send_subject,tb_send_content,tb_send_user,tb_send_date,tb_send_picture,tb_send_type,tb_send_types,tb_send_small_type,tb_send_author,tb_send_ltime) values ('".$tb_send_subject."','".$tb_send_content."','".$tb_send_user."','".$tb_send_date."','".$tb_send_picture."','".$tb_send_type."','".$tb_send_types."','".$tb_send_small_type."','".$tb_send_user."','".$tb_send_date."')",$conn)){
            mysql_query("update tb_forum_user set tb_forum_grade=tb_forum_grade+5",$conn);
                echo "<script>alert('新帖发表成功!');history.back();</script>";
            mysql_close($conn);
     }else{
              echo "<script>alert('新帖发表失败!');history.back();</script>";
              mysql_close($conn);
      }
}
if($_FILES["send_accessories][size"] > 20000000){        //判断上传附件是否超过指定的大小
    echo "<script>alert('上传文件超过指定大小！ ');history.go(-1);</script>";
    exit();
}else{
$path = './file/'.time().$_FILES['send_accessories']['name'];          //定义上传文件的路径和名称
if (move_uploaded_file($_FILES['send_accessories']['tmp_name'],$path)) { //将附件存储到指定位置
    if(mysql_query("insert into tb_forum_send(tb_send_subject,tb_send_content,tb_send_user,tb_send_date,tb_send_picture,tb_send_type,tb_send_types,tb_send_small_type,tb_send_accessories,tb_send_author,tb_send_ltime) values ('".$tb_send_subject."','".$tb_send_content."','".$tb_send_user."','".$tb_send_date."','".$tb_send_picture."','".$tb_send_type."','".$tb_send_types."','".$tb_send_small_type."','".$path."','".$tb_send_user."','".$tb_send_date."')",$conn)){
           //执行更新操作，为会员添加积分
            mysql_query("update tb_forum_user set tb_forum_grade=tb_forum_grade+5",$conn);
            echo "<script>alert('新帖发表成功!');history.back();</script>";
            mysql_close($conn);
     }else{
```



```
                echo "<script>alert('新帖发表失败!');history.back();</script>";
                mysql_close($conn);
        }
}}
?>
```

其中，需要重点说明的是，发新帖的同时，会向最后回复者 tb_send_author 和最后回复时间 tb_send_ltime 两个字段中加入发帖者的名称和发帖时间。

至此，帖子发布功能的实现过程介绍完毕，完整代码可以参考本书附带光盘中的内容。

7.4.3　帖子浏览功能实现

帖子浏览是从帖子类别的输出开始的，首先在网站的左侧框架中应用树状导航菜单输出帖子的类别，根据树状导航菜单中输出帖子的类别，设置超链接，将指定类别的帖子在右侧的框架中输出，即在 content.php 文件中输出帖子的内容。其中 left.php 中的相关程序关键代码如下：

```
<td width="84%" height="24" onClick="javascript:open_close(id_a<?php echo $myrow["tb_big_type_id"];?>);" >
    <!--设置超链接，并指定 target 的值-->
    <a href="content.php?content=<?php echo $myrow["tb_big_type_content"];?>&&content_1=<?php echo $myrows ["tb_small_type_content"];?>" target="contentFrame"><?php echo $myrow["tb_big_type_content"];?></a>
</td>
```

注意　这里的 content.php 文件是在右侧的框架中输出的内容。超链接中的 target 属性获取的是右侧框架中的链接文件的名称。设置栏目标识变量 content，代表帖子的所属专区，content_1 代表帖子的类别。

```
<td height="23">
    <a href="content.php?content=<?php echo $myrow["tb_big_type_content"];?>&&content_1=<?php echo $myrow_1["tb_small_type_content"];?>" target="contentFrame"><?php echo $myrow_1["tb_small_type_content"];?></a>
</td>
```

注意　这是根据帖子的具体类别设置超链接。有关树状导航菜单的技术可以参考本章第 7.2.1 节中的内容，这里不再复述。

然后，在 content.php 文件中输出对应类别帖子的内容。其中应用 switch 语句，根据获取的栏目标识变量$class 的不同值，分别调用不同的文件，输出不同类别中帖子的内容。content.php 文件的关键代码如下：

（1）判断论坛的所属专区和类别是否为空，如果为空则输出默认的内容，否则将输出对应专区和类别中帖子的内容。

```
<!--判断当论坛的所属专区和类别为空时，执行下面的内容-->
<?php   if($_GET["content"]=="" and $_GET["content_1"]==""){ ?>
<table><tr><td height="10"> </td></tr>
```

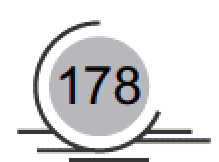
```
<tr>
  <td><?php include_once("bccd.php");?></td> <!--执行包含文件-->
</tr>
</table>
<?php }else{?>
```

（2）创建一个搜索引擎的表单，为不同的类别和专区设置超链接，并设置超链接的栏目标识变量。关键代码如下：

```
<form name="form1" method="post" action="search.php?class=搜索引擎&&content=<?php echo $_GET
[content];?>&&content_1=<?php echo $_GET[content_1];?>" onSubmit="return check_submit();">
<tr><td width="10%" height="40" rowspan="2" valign="middle">
        <a href="content.php?class=最新帖子&&content=<?php echo $_GET[content];?>&&content_1=<?php
echo $_GET[content_1];?>"><img src="images/index_7 (1).jpg" width="65" height="23" border="0"></a></td>
    <td width="10%" rowspan="2" valign="middle">
        <a href="content.php?class=精华区&&content=<?php echo $_GET[content];?>&&content_1=<?php
echo $_GET[content_1];?>"><img src="images/index_7 (2).jpg" width="55" height="23" border="0"></a></td>
    <td width="10%" rowspan="2" valign="middle">
        <a href="content.php?class=热点区&&content=<?php echo $_GET[content];?>&&content_1=<?php
echo $_GET[content_1];?>"><img src="images/index_7 (3).jpg" width="52" height="23" border="0"></a></td>
    <td width="10%" rowspan="2" valign="middle">
        <a href="content.php?class=待回复&&content=<?php echo $_GET[content];?>&&content_1=<?php
echo $_GET[content_1];?>"><img src="images/index_7 (4).jpg" width="55" height="23" border="0"></a></td>
          <td width="25%" height="38" align="right" valign="bottom">
              <input name="tb_send_subject_content" type="text" size="20" />  </td>
    <td width="25%" rowspan="2">
        <input type="image" name="imageField" src="images/index_71.jpg" /></td>
</tr>
</form>
```

（3）编写 switch 语句，根据栏目标识变量$class 的不同值，调用不同的文件。代码如下：

```
<?php
if(empty($_GET["class"]))
$class="";                                   //如果 class 没有传递值，则赋值为空
else
$class=$_GET["class"];                       //如果 class 有传递值，则将获得的 class 值赋给$class
 switch($class){
           case "最新帖子":
                include("new_forum.php");
           break;
           case "精华区":
                include("distillate.php");
           break;
           case "热点区":
                include("hotspot.php");
           break;
           case "待回复":
                include("pending.php");
```



```
				break;
			case "":
				include("new_forum.php");
				break;
		}
	}
?>
```

根据$class 变量的不同值调用不同的文件，在被调用的这些文件中，读取数据库中数据的方法都是相同的，都是根据所属的类别从数据库中读取出符合条件的数据，进行分页显示。

这里以最新帖子中调用的 new_forum.php 文件为例，对被调用文件的创建方法进行讲解。

在 new_forum.php 文件中，主要就是以超链接栏目标识中传递的变量$_GET[z"content"]和$_GET ["content_1"]为条件，从数据库中读取出符合条件的数据。

（1）输出的是所属专区中公告和置顶帖子的标题信息，并且设置超链接，链接到 send_affiche.php 和 send_forum_content.php 文件，在对应的文件中输出公告和置顶帖子的详细内容。其中，new_forum.php 文件的关键代码如下：

```
<?php
$query=mysql_query("select * from tb_forum_affiche where tb_affiche_type='$_GET[content] '");
while($myrow=mysql_fetch_array($query)){                    //输出公告内容
?>
  <tr>
    <td width="90%" colspan="4"><a href="send_affiche.php?tb_affiche_type=<?php echo $myrow [tb_affiche_type];?>&&tb_affiche_id=<?php echo $myrow["tb_affiche_id"];?>" target="_blank"><?php echo $myrow[tb_affiche_subject];?></a></td>
  </tr>
  <?php }
$query_1=mysql_query("select * from tb_forum_send where tb_send_type='1' and tb_send_small _type='".$_GET[content_1]."'");
while($myrow_1=mysql_fetch_array($query_1)){                //输出置顶内容
?>
  <tr>
    <td colspan="4" width="90%"><a href="send_forum_content.php?send_big_type=<?php echo $_GET ["content"];?>&&send_small_type=<?php echo $myrow_1["tb_send_small_type"];?>&&send_id=<?php echo $myrow_1 ["tb_send_id"];?>" target="_blank"><?php echo $myrow_1["tb_send_subject"];?></a></td>
  </tr>
  <?php }?>
```

（2）根据栏目标识传递的变量，从数据库中读取出对应的专区和类别中帖子的数据，将读出的帖子按照最后发布的时间进行倒序排列，这样即可实现最后发布的帖子可以在会员浏览相关类别文章时优先被读取，之后定义变量，实现数据的分页显示；在输出帖子的标题时，设置超链接，链接到 send_forum_content.php，在该文件中输出帖子的详细信息。程序关键代码如下：

```
<?php
if($page){
    $page_size=1;                                           //定义每页显示的条数
```

```
$query="select count(*) as total from tb_forum_send where tb_send_small_type='$_GET[content_1]'";
$result=mysql_query($query);                              //执行查询语句
$message_count=mysql_result($result,0,"total");           //获取总的记录数
$page_count=ceil($message_count/$page_size);              //计算总的页数
$offset=($page-1)*$page_size;                             //计算上一页的最后一条记录
//从数据库中读取帖子的数据，按照帖子发布的时间进行降幂排列输出
$query_2=mysql_query("select * from tb_forum_send where tb_send_small_type='$_GET[content_1]' order by
tb_send_ltime desc limit $offset, $page_size");
while($myrow_2=mysql_fetch_array($query_2)){//循环输出查询结果
?>
<tr>
    <td align="center"><img src="<?php echo $myrow_2["tb_send_picture"];?>"></td>
    <td><a href="send_forum_content.php?send_big_type=<?php echo $_GET["content"];?>&&send_small_
type=<?php echo $myrow_2["tb_send_small_type"];?>&&send_id=<?php echo $myrow_2["tb_send_id"];?>"
target="_blank"><?php echo $myrow_2["tb_send_subject"];?></a></td>
    <td><?php echo $myrow_2["tb_send_date"];?></td>
    <td><?php echo $myrow_2["tb_send_user"]."/";?>
    <?php $query_ta=mysql_query("select * from tb_forum_send where tb_send_id=$myrow_2[tb_send_id]");
    $myrow_ta=mysql_fetch_array($query_ta);
    if($myrow_ta["tb_send_types"]==1)                     //输出最后回复帖子的用户
        echo $myrow_ta["tb_send_author"];
        else
        echo $myrow_ta["tb_send_user"];
    ?> </td>
        <td><?php   $query_s=mysql_query("select * from tb_forum_restore where tb_send_id='$myrow_2[tb_
send_id]'");                                              //输出回复数以及帖子访问量
    echo mysql_num_rows($query_s);
echo "/";
$query_tn=mysql_query("select * from tb_forum_send where tb_send_id=$myrow_2[tb_send_id]");
$myrow_tn=mysql_fetch_array($query_tn);
echo $myrow_tn["tb_send_count"];
?></td>
</tr>
<?php }}?>
```

上述讲解的是帖子主题浏览的实现方法，下面介绍帖子内容浏览功能的实现。

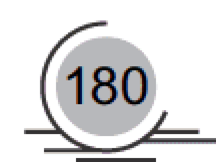
帖子内容的输出是通过上面提到的 send_forum_content.php 文件来完成的。在该文件中输出帖子的详细内容、发帖人的信息、回复帖子的内容和回复人的信息，并且还对登录用户进行了权限设置。普通用户只能浏览帖子的详细信息，不能进行其他任何操作。

会员登录后不但可以浏览帖子的详细信息，而且可以对帖子进行回复和引用，收藏帖子、发送短信以及加对方为好友；如果浏览的是当前会员自己发布或者回复的帖子，还可以对帖子进行修改、删除和结帖的操作。

管理员登录后可以执行上述会员具有的所有操作，并且还可以对帖子进行置顶的操作，这是会员所不具备的。

下面对 send_forum_content.php 文件进行分步讲解，看看各个部分的功能都是如何实现的。

（1）初始化 Session 变量，连接数据库，通过 include 语句调用包含文件，通过$_SERVER 预定义变量获取当前页面的完整连接地址，用于实现帖子收藏的功能。有关帖子收藏技术的讲解请参考 7.2.4



节中的内容。

（2）输出发帖人的信息和发布的帖子信息，并且为发送短信、加为好友、结帖、置顶、修改、删除和回复的操作设置超链接。发帖信息文件是 send_forum_content.php，其关键代码如下：

```
<!--从数据库中读取出指定帖子的发布人的信息-->
<?php
$query_1=mysql_query("select * from tb_forum_send where tb_send_id='$_GET[send_id]'",$conn);
$myrow_1=mysql_fetch_array($query_1);                    //读取指定帖子
$query_2=mysql_query("select * from tb_forum_user where tb_forum_user='$myrow_1[tb_send_user]'",$conn);
$myrow_2=mysql_fetch_array($query_2);                    //从会员信息表中读取指定的会员信息
echo "<img src='$myrow_2[tb_forum_picture]'>"."<br>";    //输出图片
echo "发帖人:";
echo $myrow_2["tb_forum_user"]."<br>";                   //输出会员名
echo "注册时间:"."<br>";
echo $myrow_2["tb_forum_date"]."<br>";                   //输出注册时间
echo "积分:";
echo $myrow_2["tb_forum_grade"]."<br>";                  //输出积分
if($_SESSION["tb_forum_user"]==true){
echo "<a href='send_mail.php?receiving_person=$myrow_2["tb_forum_user"]&&sender=$_SESSION["tb_forum_
user"]' target='_blank'><img src='images/index_8.jpg' width='76' height='24' border='0'></a>"."<br>";
echo "<a href='my_friend.php?friend=$myrow_2["tb_forum_user"]&&my=$_SESSION["tb_forum_user"]' target=
'_blank'><img src='images/index_8 (1).jpg' width='82' height='24' border='0'></a>";
}
?>
<!--从数据库中读取出指定帖子的信息-->
<?php     if($myrow_1["tb_forum_end"]!=1){ ?>            //判断帖子是否已经结帖
<a  href="end_forum.php?send_id=<?php  echo  $_GET[send_id];?>&send_user=<?php  echo  $myrow_3
["tb_send_ user"];?>">结帖</a>
<?php    }else{    echo "已结帖";  }     ?>
</td></tr>
<tr><td height="21"><?php echo $myrow_3["tb_send_content"];   ?></td></tr>
<!--为不同功能的实现设置超链接-->
<tr><td align="right">
<?php if($myrow_3["tb_send_accessories"]==true){         //判断是否存在附件
    echo  "<a  href='download.php?accessories=$myrow_3["tb_send_accessories"]'>附件</a>";}?>  
  
    <a href="permute_send.php?permute_id=<?php echo $myrow_3["tb_send_id"];?>">置顶</a>、
    <a href="recompose_send.php?recompose_id=<?php echo $myrow_3["tb_send_id"];?>&&recompose_user=
<?php echo $myrow_3["tb_send_user"];?>">修改</a>、
    <a href="delete_send.php?delete_id=<?php echo $myrow_3["tb_send_id"];?>&&delete_user=<?php echo
$myrow_3["tb_send_user"];?>">删除</a>、
    <a  href="send_forum_content.php?send_big_type=<?php  echo  $_GET  ["send_big  _type"];?>&&send_
small_type=<?php echo $_GET["send_small_type"];?>&&send_id=<?php echo $_GET["send_id"];?>#bottom">
回复</a></td>
  </tr>
```

（3）输出与该帖子相关的回复帖子的信息，以及回复人的信息，同样也为发送短信、加为好友、引用、修改和删除操作设置了超链接。其中在输出回复帖子的内容时，还对回帖内容进行判断，判断

该帖子是否被管理员屏蔽。

实现的方法与第（2）步相同，这里不再赘述，完整代码请参考本书附带光盘中的内容。

（4）积分排行功能的实现主要就是从会员信息表中读取会员的积分数据，并且按照降幂排列，输出积分最高的前 10 名用户。程序代码如下：

```
<?php
$sql=mysql_query("select tb_forum_user,tb_forum_grade from tb_forum_user order by tb_forum_grade desc limit 10");
                                        //从数据库中读取出积分最高的 10 个用户
while($myrow=mysql_fetch_array($sql)){          //循环输出用户姓名和积分
?>
<tr><td width="45%" height="19" align="right">
     <a href="person_data.php?person_id=<?php echo $myrow[tb_forum_user];?>"><?php echo $myrow
[tb_forum_user];?></a> </td>
   <td width="55%" align="left">——<?php echo $myrow[tb_forum_grade];?></td>
</tr>
<?php }?>
```

（5）send_forum_content.php 文件的第 5 部分是一个 form 表单，用于提交回复帖子的信息。有关回复帖子的操作将在 7.4.4 节中做详细介绍。

7.4.4 帖子回复功能实现

帖子回复中提交的 form 表单存储在 7.4.3 节中介绍的 send_forum_content.php 文件中。

在这个 form 表单中，将回复主题、附件、回复内容、发布帖子 id 和回复人信息都提交到 send_forum_content_ok.php 文件中进行处理，完成帖子的回复。

在对回复内容进行编辑时还应用了 UBB 技术，实现对回复的内容进行编辑，并且还对回复内容的字节数进行了控制。

UBB 技术的实现是通过 UBBCode.js 文件来完成的，该文件存储于根目录下的 js 文件夹中；限制和统计回复内容字节数的方法是通过 text.js 文件来完成的，该文件同样存储于根目录下的 js 文件夹中。form 表单存在于 send_forum_content.php 文件中，其关键代码如下：

```
<form action="send_forum_content_ok.php" method="post" enctype="multipart/form-data"  name="myform">
<tr><a name="bottom" id="bottom"></a><!--定义命名锚记-->
<!--输出引用的回帖的标题-->
   <td width="641"><input name="restore_subject" type="text" id="restore_subject" value="
<?php
if($_GET["cite"]==true){
   //从数据库中获取回复帖子的标题
   $query=mysql_query("select * from tb_forum_restore where tb_restore_id='$_GET[cite]'");
   $result=mysql_fetch_array($query);
echo "摘自（".$result["tb_restore_user"]."）：".$result[tb_restore_subject];     //输出引用标题
}
?>" size="80">
   <input type="hidden" name="tag" value="<?php echo $myrow_3["tb_forum_end"];?>"></td>
</tr>
```

```
<tr><td height="30" align="right">文字编程区：</td>
    <td width="641">
        <img src="images/UBB/B.gif" width="21" height="20" onClick="bold()"> 
        <img src="images/UBB/I.gif" width="21" height="20" onClick="italicize()">字体
        <select name="font" class="wenbenkuang" id="font" onChange=" showfont(this.options [this.
selectedIndex].value)">
            <option value="宋体" selected>宋体</option>
            <option value="黑体">黑体</option>
        </select></td>
</tr>
<tr>
    <td align="right">文章内容：</td>
    <td width="641"><!--统计回复内容，输出引用的回复内容-->
        <textarea name="file" cols="80" rows="10" id="file" onKeyDown="countstrbyte(this.form.file,this.form.
total,this.form.used,this.form.remain);" onKeyUp="countstrbyte (this.form.file,this.form.total,this.form.used,this.
form.remain);">
<?php
if($_GET["cite"]==true){
    //输出回复帖子的内容
    $query=mysql_query("select * from tb_forum_restore where tb_restore_id='$_GET[cite]'");
    $result=mysql_fetch_array($query);
    echo $result[tb_restore_content];                    //输出引用的回复内容
}
?>
        </textarea>
        <input type="hidden" name="tb_send_id" value="<?php echo $_GET["send_id"];?>">
        <input type="hidden" name="tb_restore_user" value="<?php echo $_SESSION["tb_forum_user"];?>">
</td>
</tr>
<tr>
    <td height="25" colspan="2">        
        <input name="submit" type="submit" id="submit" value="提交" onClick="return check();">
    最大字节数：<input type="text" name="total" disabled="disabled" class="textbox" id="total" value="500"
size="5">
    输入：<input type="text" name="used" disabled="disabled" class="textbox"  id="used" value="0" size="5">
    <input name="reset" type="reset" id="reset" value="重写"></td>
</tr>
</form>
```

至此，回复帖子的提交文件的内容讲解完毕，接下来介绍回复帖子的处理页 send_forum_content_ok.php 文件。

在该文件中实现对回复帖子中提交的数据进行存储，并且更新帖子的回复次数，将最后回复的会员名和回复时间更新到 tb_forum_send 表中的 tb_send_author 和 tb_send_ltime 字段下，实现在论坛浏览页面显示最后回复该帖子的会员名，并在浏览页面将最新回复的帖子移置所有帖子最顶端，同时将发布帖子数据表（tb_forum_send）中的 tb_send_types 字段更新为 1，表明该帖子已经有回帖。

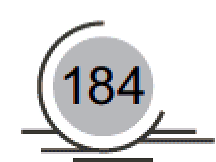
在 send_forum_content_ok.php 文件中，首先获取到 form 表单中提交的数据，然后判断回复的内容中是否包含附件，如果不存在附件，则直接将获取的数据添加到指定的数据表中，并且更新帖子的回复次数和发布帖子数据表中 tb_send_types 字段的值为 1。

如果存在附件，而附件的大小超过上传文件大小的限制，则将给出提示信息“上传文件超过指定

大小！”。

如果存在附件，并且在指定的范围之内，则先将该附件存储到服务器中指定的文件夹下，然后再将附件在服务器中的存储路径和其他数据一起存储到指定的数据表中，同样也更新帖子的回复次数和发布帖子数据表中 tb_send_types 字段的值为 1。该过程是在 send_forum_content_ok.php 文件中实现的，相关程序代码如下：

```
<?php SESSION_start(); include_once("conn/conn.php");
if($_SESSION["tb_forum_user"]==true){                    //判断是否是正确登录
$tb_restore_subject=$_POST["restore_subject"];           //获取回复帖子的主题
$tb_restore_content=$_POST["file"];                      //获取上传的附件
$tb_restore_user=$_POST["tb_restore_user"];              //获取回复人
$tb_send_id=$_POST["tb_send_id"];                        //获取要回复帖子的 id
$tb_restore_date=date("Y-m-d H:i:s");                    //定义回复时间
$tb_send_date=date("Y-m-d H:i:s");                       //定义最后回复时间
$cite=$_POST["cite"];                                    //定义引用值
if($_FILES["restore_accessories"]["size"]==0){           //判断是否有附件上传
if(mysql_query("insert into tb_forum_restore (tb_restore_subject,tb_ restore_content,tb_restore_ user,tb_
send_id,tb_restore_date) values ('".$tb_restore_ subject."','".$tb_ restore_content."','". $tb_restore_user."','".
$tb_send_id."','".$tb_restore_date."')",$conn)){
//更新最后发布者和最后回复时间的操作
mysql_query("update tb_forum_send set tb_forum_send.tb_send_author='$tb_restore_user' where tb_send_id
= $tb_send_id");
 mysql_query("update tb_forum_send set tb_forum_send.tb_send_ltime='$tb_send_date' where tb_send_id =
$tb_send_id");
//执行更新回复次数的操作
 mysql_query("update tb_forum_restore set tb_forum_counts=tb_forum_counts+1",$conn);
 mysql_query("update tb_forum_send set tb_send_types=1 where tb_send_id='$tb_send_id'",$conn);
          echo "<script>alert('回复成功!');history.back();</script>";
 mysql_close($conn);
 }else{
   echo "<script>alert('回复失败!');history.back();</script>";
   mysql_close($conn);
  }
}
if($_FILES["restore_accessories"]["size"] > 20000000){ //判断上传附件是否超过规定文件的大小
    echo "<script>alert('上传文件超过指定大小！');history.go(-1);</script>";
    exit();
}else{
$path = './file/'.time().$_FILES['restore_accessories']['name'];    //定义上传文件名称和存储的路径
//将附件存储到服务器指定的文件夹下
if (move_uploaded_file($_FILES['restore_accessories']['tmp_name'],$path)) { //执行存储操作
    if(mysql_query("insert into tb_forum_restore (tb_restore_subject, tb_restore_content,tb_restore_ user,tb_
send_id,tb_restore_date,tb_restore_accessories) values ('".$tb_restore_ subject."','".$tb_restore_ content."','".$tb_
restore_ user."','".$tb_send_id."','".$tb_restore_date."','".$path."')",
$conn)){                                                              //执行添加操作
          mysql_query("update tb_forum_send set tb_forum_send.tb_send_author='$tb_restore_user' where
tb_send_id = $tb_send_id");
          mysql_query("update tb_forum_send set tb_forum_send.tb_send_ltime='$tb_send_date' where tb_
```



```
send_id = $tb_send_id");
mysql_query("update tb_forum_restore set tb_forum_counts=tb_forum_counts+1",$conn);
            mysql_query("update tb_forum_send set tb_send_types=1 where tb_send_id='$tb_send_id'",$conn);
            echo "<script>alert('回复成功!');history.back();</script>";
            mysql_close($conn);
      }else{
              echo "<script>alert('回复失败!');history.back();</script>";
              mysql_close($conn);
       }
}}
}else{
    echo "<script>alert('对不起，您不可以回复帖子，请先登录到本站');history.back();</script>";
}
?>
```

在论坛显示页面中，显示的各个主题内容是按照最后回复的次序倒序排序的。这样可以使用户最后回复的帖子始终处于最上方，使其他用户能够第一时间浏览到这个帖子以及回复内容。这一设定本来是为了方便用户能够及时发现最新的帖子，但是同时也会导致一些问题，会有一些会员回复很久以前的帖子，这样老主题就会显示在页面最上方，这种恶意回复帖子的行为被成为“掘墓”。为了防止这种用户恶意行为，解决该问题的办法是在该提交处理页也添加一段代码来防止这种“掘墓”行为，相关代码如下：

```
$query_k=mysql_query("select * from tb_forum_send where tb_send_id='$tb_send_id'"); //从数据库中提取发帖时间
$myrow_k=mysql_fetch_array($query_k);
$newtime=substr($myrow_k["tb_send_date"],0,10);                    //把发帖的时间截取为“年-月-日”的形式
$array=explode("-",$newtime);                                      //用“-”做分隔符截取时间
$year=date("Y");                                                   //获取当前时间
$month=date("m");
$day=date("d");
$sendtime=($year-$array[0]-1)*365+abs(($day-$array[1]))*30+abs($day-$array[2]);   //计算已发帖时间
if($sendtime>180)                                                  //如果发帖时间已经超过 180 天
echo "<script>alert('该帖发布时间距今已超过半年，不能回复!');history.back();</script>"; //则不能进行回复
else{
//执行回复代码
}
```

至此，帖子回复功能讲解完毕，详细的程序代码可以参看本书附带光盘中的内容。

7.5　帖 子 搜 索

7.5.1　帖子搜索概述

帖子搜索是指在论坛中按照指定的关键字，从论坛发布的帖子和回复的帖子中查询出符合条件的数据。

基于论坛较高的时效性，一些大中型热门论坛每天会被大量的用户访问、发帖、回复，这同样也给浏览者在帖子查找上带来诸多不便。为了查找相关内容而浏览每一页上的所有帖子是不现实的，而帖子搜索功能即可满足浏览者的要求，迅速地查找到与浏览者提交的关键字有关的内容。

在论坛模块中，帖子搜索从 content.php 文件中设置的帖子搜索文本框开始（如图 7.28 所示），将要搜索的关键字提交到 search.php 文件中，在该文件中执行模糊查询，并将查询的结果输出，如图 7.29 所示。

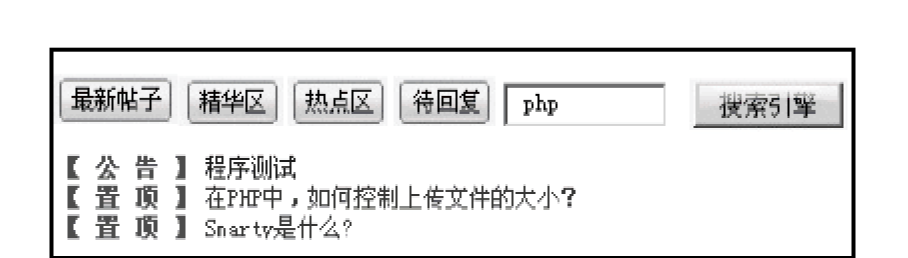


图 7.28　提交搜索关键字



图 7.29　帖子搜索的结果

7.5.2　帖子搜索功能实现

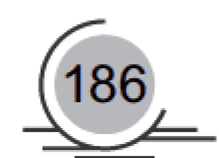
帖子搜索通过论坛首页的搜索文本框提交关键字，对站内的所有帖子执行搜索操作，将搜索得到的帖子标题名称、发帖人、回复次数等信息以超链接的形式在搜索显示页中输出。另外，一些与关键字有关的帖子，也会以超链接形式在发布的帖子下方输出。帖子搜索的实现过程如下：

（1）在 content.php 文件中，首先创建一个 form 表单，提交帖子搜索的关键字，将关键字提交到 search.php 文件中。content.php 文件中的关键代码如下：

```
<form name="form1" method="post" action="search.php?class=搜索引擎&&content=<?php echo $_GET
["content"];?> &&content_1=<?php echo $_GET["content_1"];?>" onSubmit="return check_submit();">
<tr>
    <td><input name="tb_send_subject_content" type="text" size="20" />  </td>
    <td width="25%" rowspan="2"><input type="image" name="imageField" src="images/index_71.jpg" /></td>
  </tr>
</form>
```

（2）创建 search.php 文件，根据表单中提交的关键字，分别在发布帖子和回复帖子中执行模糊查询，将查询的结果以分页的形式输出到页面中。其中模糊查询的关键代码如下：

```
<?php session_start(); include("conn/conn.php");            //初始化 Session 变量，连接数据库
if($page==""){ $page=1; }                                    //判断变量的值是否为空，用于分页显示
if($pages==""){ $pages=1; }                                  //判断变量的值是否为空，用于分页显示
if($link_type==""){ $link_type=0; }
if($link_types==""){ $link_types=0; }
$content=$_GET["content"];                                   //获取帖子的类型
```



```
$content_1=$_GET["content_1"];                              //获取帖子的类别
//从发布的帖子中查询
$query_6=mysql_query("select * from tb_forum_send where tb_send_subject like '%$tb_send_subject_content%'
or tb_send_content like '%$tb_send_subject_content%'");
//从回复的帖子中搜索
$query_7=mysql_query("select * from tb_forum_restore where tb_restore_subject like '%$tb_send_subject_
content%' or tb_restore_content like '%$tb_send_subject_content%'");
//统计查询的结果
if(mysql_num_rows($query_6)>0 or mysql_num_rows($query_7)>0 ){
    if($page){                                              //定义分页的变量
        $page_size=10;                                      //定义每页显示的数量
        $query="select count(*) as total from tb_forum_send where tb_send_subject like '%$tb_send_subject_
content%' or tb_send_content like '%$tb_send_subject_content%'";
        $result=mysql_query($query);                        //执行查询语句
        $message_count=mysql_result($result,0,"total");     //获取查询结果
        $page_count=ceil($message_count/$page_size);        //计算共有几页
        $offset=($page-1)*$page_size;                       //获取上一页的最后一条记录
        $query_2=mysql_query("select * from tb_forum_send where tb_send_subject like '%$tb_send_
subject_ content%' or tb_send_content like '%$tb_send_subject_content%' limit $offset, $page_size");
?>
```

至此，帖子搜索功能讲解完毕，有关查询结果的分页输出，这里不做介绍，完整代码可以参考本书附带光盘中的内容。

7.6　帖　子　管　理

7.6.1　帖子管理概述

对论坛帖子的管理，已成为论坛维护必不可少的一部分，帖子管理甚至可以直接影响到用户的访问和回帖数量， 其中帖子管理包括帖子的结帖功能、优秀帖子的置顶操作、设置帖子的类别以及顶帖管理的实现等。

结帖功能是对会员自己发布的帖子进行操作，当获取到满意的答案之后，就可以对帖子进行结帖操作，一旦结帖之后就不可以再对该帖进行回复。其运行结果如图 7.30 所示。



图 7.30　结帖功能的运行结果

结帖功能避免了在论坛中对一个帖子无休止地回复，浪费系统资源，同时也确保了论坛中帖子的规范性。

帖子管理的另一功能是帖子的分类输出。在论坛中，根据帖子的发布时间、帖子内容的特殊性以及受关注的程度，还有帖子是否有人回复等，对帖子进行了分类处理，分为最新帖子、精华区、热点区和待回复等几个类别。其运行结果如图 7.31 所示。



图 7.31　帖子的分类输出

顶帖管理是针对管理员的帖子置顶权限而设置的，因为不可能将某个帖子永远置顶，所以创建了顶帖管理这个功能。该功能存在于论坛的后台管理中，实现帖子置顶和取消置顶的操作。其运行结果如图 7.32 所示。

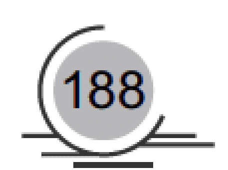


图 7.32　顶帖管理功能的运行结果

7.6.2　结帖功能实现

论坛的管理员也具备这个权限，可以根据帖子的回复情况，在确定已经有满意答案的情况下，而帖子发布人又没有进行结帖操作的，可以由管理员来执行这项操作。管理员的结帖操作是在论坛后台管理的帖子管理中完成的。

帖子是否已经结帖是根据帖子在数据表中 tb_forum_end 字段的值来判断的，如果字段的值为 1，



则说明帖子已经结帖，否则没有结帖。所以结帖操作就是将指定帖子在数据表中的 tb_forum_end 字段的值更新为 1。

在论坛模块中，结帖操作是通过在 send_forum_content.php 文件中设置的一个“结帖”的超链接来执行的。通过“结帖”这个超链接，链接到 end_forum.php 文件，在这个文件中根据传递的 id 值，执行更新指定帖子 tb_forum_end 字段值的操作。

在 send_forum_content.php 文件中设置 “结帖”超链接，并且根据 tb_forum_end 字段的值来判断输出的内容。关键代码如下：

```
<?php
if($myrow_1["tb_forum_end"]!=1){          //判断如果字段的值不为 1，则输出下面的链接
?>
<a href="end_forum.php?send_id=<?php echo $_GET[send_id];?>&send_user=<?php echo $myrow_3[tb_send_user];?>">结帖</a> <!--设置结帖操作的超链接-->
<?php
}else{
echo "已结帖";
}
?>
```

在 end_forum.php 文件中，执行结帖的操作，以$send_id 变量传递的帖子 id 值为依据。程序代码如下：

```
<?php session_start(); include("conn/conn.php");
if($_GET["send_id"]==true and $_GET["send_user"]==$_SESSION["tb_forum_user"]){
    $result=mysql_query("update tb_forum_send set tb_forum_end='1' where tb_send_id='".$_GET[send_id]."'");
    if($result==true){
        echo "<script>alert('结帖激活!'); history.back();</script>";
    }
}else{
        echo "<script>alert('您不具备该权限!'); history.back();</script>";
}
?>
```

7.6.3　设置帖子类别

精华区、热点区和待回复这 3 个类别的实现方法相同，都是根据数据库中帖子指定的字段值进行判断，精华区根据字段 tb_send_type_distillate 的值判断；热点区根据字段 tb_send_ type_ hotspot 的值判断；而待回复则根据字段 tb_send_types 的值判断。这里以精华区帖子的输出为例，该输出文件 distillate.php 的关键代码如下：

```
<?php
if($page){                                              //实现精华区帖子的分页输出
    $page_size=10;                                      //每页显示 10 条记录
    //以 tb_send_type_distillate 字段的值是否为 1 为条件，如果为 1 则是精华帖子，否则不是
    $query="select count(*) as total from tb_forum_send where tb_send_small_type='$_GET[content_1]' and
tb_send_type_distillate=1";
```

```
        $result=mysql_query($query);
            $message_count=mysql_result($result,0,"total");
            $page_count=ceil($message_count/$page_size);
            $offset=($page-1)*$page_size;
        $query_2=mysql_query("select * from tb_forum_send where tb_send_small_type='$content_1' and
tb_send_type_ distillate='1' limit $offset, $page_size");
        while($myrow_2=mysql_fetch_array($query_2)){
?>
```

上述介绍的是如何从数据库中获取到指定类别的帖子，下面讲解这些帖子的类别是如何设置的。

最新帖子不需要任何设置，只要是帖子发布之后，自动会生成一个 id 值，根据 id 值自动可以读取到最新的帖子。

而精华区、热点区和待回复则都需要设置。其中，设置精华区和热点区帖子的方法相同，都是在论坛的后台管理中进行操作，通过 form 表单创建复选框，将指定帖子的 tb_send_type_distillate 或者 tb_send_type_hotspot 字段的值设置为 1，运行结果如图 7.33 所示。



图 7.33　设置帖子的类别

帖子类别的设置操作通过两个文件完成：一个是 update_forum.php，用于提交要设置类别帖子的 id；另一个是 update_forum_ok.php，根据提交的 id 值执行设置帖子类别的操作。

在 update_forum.php 文件中，首先创建一个 form 表单，从数据库中读取帖子的数据，并且为每个帖子设置一个复选框，复选框的值是帖子的 id；再分别创建精华帖和热点帖子的“提交”按钮，同时也创建取消帖子类别的按钮；最后将数据提交到 update_forum_ok.php 文件中。其中，update_forum.php 文件的关键代码如下：

```
<form name="form1" method="post" action="update_forum_ok.php">
  <tr>
    <td height="25" align="center" class="STYLE1"><input name="<?php echo $myrow["tb_send_id"];?>"
type=" checkbox" value="<?php echo $myrow["tb_send_id"];?>"></td>
    <td align="center" class="STYLE1"><?php echo $myrow["tb_send_type_distillate"];?></td>
    <td align="center" class="STYLE1"><?php echo $myrow["tb_send_type_hotspot"];?></td>
  </tr>
  <tr>
```

```
    <td align="center"><input type="submit" name="Submit" value="精华帖">
        <input type="submit" name="Submit3" value="取消"></td>
    <td align="center"><input type="submit" name="Submit2" value="热门帖">
        <input type="submit" name="Submit4" value="取消"></td>
    <td colspan="3" align="center"><input type="submit" name="Submit5" value="结帖">
        <input type="submit" name="Submit6" value="取消"></td>
    </tr>
 </form>
```

在 update_forum_ok.php 文件中，根据表单中提交的帖子的 id 值，通过 while 语句和 list()函数，循环读取表单中提交的帖子的 id 值，执行设置帖子类别和取消帖子类别的操作。关键代码如下：

```
<?php session_start(); include("conn/conn.php");
if(!empty($_POST["Submit"]))                          //判断从表单获取的 Submit 是否为空
$Submit=$_POST["Submit"];                             //如果不为空，则将值赋给变量$Sumbit
if(!empty($_POST["Submit2"]))
$Submit2=$_POST["Submit2"];
if(!empty($_POST["Submit3"]))
$Submit3=$_POST["Submit3"];
if(!empty($_POST["Submit4"]))
$Submit4=$_POST["Submit4"];
if(!empty($_POST["Submit5"]))
$Submit5=$_POST["Submit5"];
if(!empty($_POST["Submit6"]))
$Submit6=$_POST["Submit6"];
if($Submit=="精华帖"){                                 //判断当表单提交的值等于精华帖时执行下面的操作
    while(list($name,$value)=each($_POST)){           //通过 list()函数获取表单提交的值，并循环输出内容
        $result=mysql_query("update tb_forum_send set tb_send_type_distillate='1' where tb_send_id=
'".$name."'");                                        //执行设置精华帖的操作
    if($result==true){
        echo "<script>alert('精华帖激活成功!'); window.location.href='index.php?title=帖子管理';</script>";}}
}
if($Submit3=="取消"){                                  //执行取消精华帖的设置
    while(list($name,$value)=each($_POST)){
        $result=mysql_query("update tb_forum_send set tb_send_type_distillate='0' where tb_send_id=
'".$name."'");
    if($result==true){
        echo "<script>alert('精华帖取消!'); window.location.href='index.php?title=帖子管理';</script>";}}
}?>
```

注意 在设置帖子类别的过程中，使用的是批量更新技术，其主要通过 while 循环语句和 list()、each()函数来完成。

（1）each()函数返回数组中当前指针位置的键名和对应的值，并向前移动数组指针。键值对被返回为 4 个单元的数组，键名为 0、1、key 和 value。单元 0 和 key 包含数组单元的键名，1 和 value 包含数据，如果内部指针越过了数组的末端，则函数返回 false。该函数的语法如下：

```
array each ( array array)
```

参数 array 为输入的数组。

（2）list()函数把数组中的值赋给一些变量。与 array()函数类似，不是真正的函数，而是语言结构。list()函数仅能用于数字索引的数组，并且数字索引从 0 开始。该函数的语法如下：

```
void list ( mixed…)
```

参数 mixed 为被赋值的变量名称。

（3）值得一提的是，while(list($name,$value)=each($_POST))这条语句不能在同一文件中被重复利用，即在执行完该操作循环体内容后，不能通过另一条与该语句并行且相同的循环语句再次执行，这样可能会导致一个非预期的效果，代码如下：

```
<?php
while(list($name,$value)=each($_POST)){
echo "这条语句会被输出";                    //该语句会被执行
}
while(list($name,$value)=each($_POST)){
echo "这条语句不会被输出";                  //该语句不会被执行
}
?>
```

而待回复则是在回复帖子的操作中完成的，当回复帖子提交成功后，执行更新回复帖子中字段 tb_send_types 的值为 1，表明该帖子已经有回复，有关程序代码可以参考第 7.4.4 节中的内容，这里不再赘述。

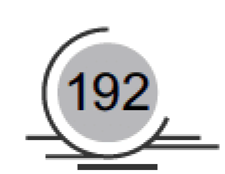
7.6.4 顶帖管理功能的实现

顶帖管理是针对管理员的帖子置顶权限而设置的，因为不可能将某个帖子永远置顶，所以创建了顶帖管理这个功能。该功能存在于论坛的后台管理中，实现帖子置顶和取消置顶的操作。其运行结果如图 7.34 所示。



图 7.34　顶帖管理功能的运行结果

顶帖管理功能的实现应用了两个文件：一个是 permute_admin.php，用于从数据库中读取出置顶帖子的数据，进行分页输出，并且创建 foum 表单，为每个帖子设置一个下拉列表框，实现帖子置顶和取



消置顶的操作。

其中，permute_admin.php 文件的关键代码如下：

```
<?php
    if($page){
        $page_size=5;                                                  //每页显示 5 条记录
        $query="select count(*) as total from tb_forum_send where tb_send_type=1";
        $result=mysql_query($query);                                   //执行查询操作
        $message_count=mysql_result($result,0,"total");               //获取总的记录数
        $page_count=ceil($message_count/$page_size);                  //获取总的页数
        $offset=($page-1)*$page_size;
        $query=mysql_query("select * from tb_forum_send where tb_send_type=1 order by tb_send_id desc limit
$offset, $page_size");
while($myrow=mysql_fetch_array($query)){                              //循环输出查询结果
?>
  <tr>
    <td align="center">
<form action="update_permute.php?update_id=<?php echo $myrow["tb_send_id"];?>" method="post" name="
form1" class="STYLE1">
        <select name="tb_send_type" id="tb_send_type">
            <option value="1">置顶</option>
            <option value="0">取消</option>
        </select>
        <input type="submit" name="Submit" value="执行">
</form></td>
  </tr>
<?php }}?>
```

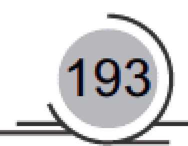
另一个是 update_permute.php，在该文件中，根据表单中提交的值和帖子 id 的值，将数据库中 tb_send_type 字段的值改为提交的相关数值，来实现帖子置顶和取消置顶的操作。关键代码如下：

```
<?php    include("../conn/conn.php");
$update_id=$_GET["update_id"];           //获取帖子的 ID 值
//执行帖子置顶或者取消置顶的操作
$query=mysql_query("update tb_forum_send set tb_send_type='$_POST[tb_send_type]' where tb_send_id='
$update_id'");
if($query==true){
    echo "<script>alert('更新成功!');history.back();</script>";
}else{
    echo "<script>alert('更新失败!');history.back();</script>";}
?>
```

7.7　个人信息管理

7.7.1　个人信息管理概述

作为论坛模块，登录用户个人信息的显示也是必不可少的元素之一。在整个系统中，用户不但可



以通过个人邮箱收发站内邮件，而且可以对个人的站内邮件进行管理。此外，登录用户还可以通过帖子上所显示的个人信息添加站内其他会员为好友，并通过个人信息管理模块管理个人好友名单。

用户的个人信息管理主要通过用户个人显示信息模块实现，如图 7.35 所示。



图 7.35　用户个人信息显示

用户登录成功后，可以看到登录用户的个人信息，单击“我的好友”或“我的信箱”，即可跳转至相应的管理页面。另外，该模块还包含了个人信息更改、我参与的帖子、我的收藏等管理超链接。由于篇幅限制，这里只对我的信箱和我的好友做详细讲解，其他相关程序请参见光盘中的内容。

7.7.2　我的信箱管理

创建我的信箱从第 7.4.3 节 send_forum_content.php 文件中设置的超链接开始。我的信箱模块链接的是 send_mail.php 文件，在该文件中完成我的信箱模块的操作。其运行结果如图 7.36 所示。

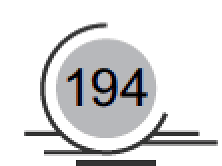


图 7.36　我的信箱模块的运行结果

我的信箱模块中主要包括写信、收件箱和发件箱 3 个功能。这 3 个功能都在 send_mail.php 文件中完成。

在 send_mail.php 文件中，应用 switch 语句，根据栏目标识的变量值，实现不同功能之间的切换输出。该实现过程的关键代码如下：

```
<table width="812" height="65" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td><?php echo $_GET[sender];?>您好：您现在有
<?php
    $query=mysql_query("select * from tb_mail_box where tb_receiving_person='$_GET[sender]' and tb_mail_
type=''");
```





```
        $myrow=mysql_num_rows($query);
        echo $myrow;
?>条未读信息！</td>
            </tr>
            <tr>
              <td><a href="send_mail.php?sender=<?php echo $_GET[sender];?>&&mails=收件箱">收件箱</a></td>
              <td><a href="send_mail.php?sender=<?php echo $_GET[sender];?>&&mails=发件箱">发件箱</a></td>
              <td><a href="send_mail.php?sender=<?php echo $_GET[sender];?>&&mails=写信">写信</a></td>
            </tr>
</table>
<?php
        switch($mails){
            case "":
                    include("write_mail.php");
            break;
            case "写信":
                    include("write_mail.php");
            break;
            case "收件箱":
                    include("browse_mail.php");
            break;
            case "发件箱":
                    include("browse_send_mail.php");
            break;
        }
?>
```

（1）写信通过 write_mail.php 和 write_mail_ok.php 文件完成。通过 write_mail.php 文件来创建 form 表单，提交发送信息的内容。通过 write_mail_ok.php 文件来对表单中提交的内容进行处理，并且将发送信息存储到指定的数据表中作为发送记录。

（2）收信通过 browse_mail.php、browse_mail_content.php 和 delete_mail.php3 个文件来完成。通过 browse_mail.php 文件从数据库中读取出收到信息的内容，将信息内容进行分页输出，并且设置超链接，链接到 browse_mail_content.php 文件。在 browse_mail_content.php 文件中查看信息的详细内容。通过 delete_mail.php 文件，实现对收信箱中的信息进行管理，按照不同类别的信件可以执行不同的操作，如系统邮件只能进行删除和查看操作，普通邮件则可以进行查看、回复、删除操作，好友验证邮件可以执行将发送验证的会员加为好友操作。其运行结果如图 7.37～图 7.39 所示。



图 7.37　系统邮件查看



图 7.38　验证邮件查看

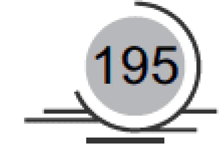


图 7.39　普通邮件查看

判别邮件类型是通过 tb_mail_box 表中的 yanzheng 字段来决定的，通过相关参数更新数据库的相关类别，然后通过返回的相关字段名称来判断站内邮件的性质，最后通过判断的结果输出对应的邮件类别，该过程是通过 browse_mail.php 文件实现的，其相关程序代码如下：

```
<?php session_start();include("conn/conn.php");
if(!empty($_GET["mail_id"])){
	$query=mysql_query("update tb_mail_box set tb_mail_type=1 where tb_mail_id='$_GET[mail_id]'");
}
?>
//省略部分代码
<table width="950" height="185" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#F7F7FF">
<?php	$query=mysql_query("select * from tb_mail_box where tb_mail_id='$_GET[mail_id]'");
		$myrow=mysql_fetch_array($query);
?>
  <tr>
    <td width="129"> </td>
    <td height="35" colspan="2" bgcolor="#FFFFFF"><div class="ttstyle">
	<div class="title">
	<span class="fontcolor">
	<?php
	if($myrow["yanzheng"]==1)				//当该字段值为 1 时为验证信息
	echo "验证信息:";
	if($myrow["yanzheng"]==2)				//当该字段值为 2 时为普通邮件
	echo "邮件标题:";
	if($myrow["yanzheng"]==3)				//当该字段值为 3 时为已验证邮件
	echo "已验证邮件";
	if($myrow["yanzheng"]==0)				//当该字段值为 0 时为系统邮件
	echo "系统邮件:";
	?>
```

根据获取数据库中相应的字段内容判断邮件类型后，执行输出操作，当判断邮件类型为普通邮件时，输出“回复”按钮，并添加隐藏域将邮件的发送者和标题等信息作为隐藏域传递到写信操作页面；当判断邮件的类型为好友验证邮件时，输出“加为好友”按钮，将自己的名称和发送请求者的会员名称作为隐藏域传递给 add_friend.php 页进行处理；如果判断邮件类型为系统邮件或者已验证邮件时，则没有执行选项输出。程序相关代码如下：

```
<?php
	if($myrow["yanzheng"]==1) {				//如果信件为好友验证信件
```

```
$tb_mail_id=$_GET["mail_id"];
$tb_friend=$myrow["tb_mail_sender"];                    //发送请求者
$tb_my=$myrow["tb_receiving_person"];                   //被请求认证者
echo "<form action=\"add_friend.php\" method=\"post\" >";
echo "<input type=\"hidden\" name=\"tb_friend\" value=\"".$tb_friend."\" >";
echo "<input type=\"hidden\" name=\"tb_my\" value=\"".$tb_my."\" >";
echo "<input type=\"hidden\" name=\"tb_mail_id\" value=\"".$tb_mail_id."\" >";
echo "<input input class=\"inputsty\" type=\"submit\" value=\"加为好友\">";
echo "</form>";
}
if($myrow["yanzheng"]==2) {                             //如果信件为普通邮件
$receiver=$myrow["tb_mail_sender"];                     //转换发信者作为收信人
$sender=$myrow["tb_receiving_person"];                  //转换收信者转变为发信人
$include=$myrow["tb_mail_content"];                     //信件内容
$subject=$myrow["tb_mail_subject"];
echo "<form action=\"send_mail.php?sender=".$sender."&&mails=写信\" method=\"post\" >";
echo "<input type=\"hidden\" name=\"receiver\" value=\"".$receiver."\" >";    //添加隐藏域
echo "<input type=\"hidden\" name=\"sender\" value=\"".$sender."\" >";
echo "<input type=\"hidden\" name=\"include\" value=\"".$include."\" >";
echo "<input type=\"hidden\" name=\"subject\" value=\"".$subject."\" >";
echo "<input class=\"inputsty\"type=\"submit\" value=\"添加回复\">";
echo "</form>";
}
?>
```

当判断邮件为验证邮件后，单击“加为好友”按钮，将当前登录的会员名称和发送请求的会员名称作为传递值传递到 add_friend.php 页面进行处理，添加成功后，自动向申请好友的会员发送系统邮件，告之对方已经将其加为好友。同时也将验证信件字段更改为“已验证”状态。其中，添加好友处理页 add_friend.php 的程序代码如下：

```
<?php
include("conn/conn.php");
$tb_friend=$_POST["tb_friend"];
$tb_my=$_POST["tb_my"];
$tb_mail_id=$_POST["tb_mail_id"];
$tb_date=date("Y-m-d");
$querys=mysql_query("select * from tb_forum_user where tb_forum_user.tb_forum_user='$tb_friend'");
if(mysql_num_rows($querys)>0) {
//通过验证后字段改为 3
$queryss=mysql_query("update tb_mail_box set yanzheng = 3 where tb_mail_id =$tb_mail_id");
//向发送请求的用户发送系统通知
        $tb_mail_subject=$tb_my."已经通过好友申请验证";
$tb_mail_content=$tb_my."已经通过了您的好友申请!";
        $querys=mysql_query("insert  into  tb_my_friend(tb_my,tb_friend,tb_date)values('$tb_my','$tb_friend','$tb_
date')");
        $query=mysql_query("insert  into  tb_mail_box(tb_receiving_person,tb_mail_subject,tb_mail_content,tb_mail_
sender,tb_mail_date,yanzheng )values('$tb_friend','$tb_mail_subject','$tb_mail_content','系统信息','$tb_date',0)");
                if($query==true){
                        echo "<script>alert('验证通过!已将对方加为好友');history.back();</script>";
```

```
                }
        }else{
            echo "<script>alert('对不起，不存在该用户!');history.back();</script>";
        }
?>
```

删除指定信息的操作可以参考 7.2.8 节中的内容，这里不再赘述。

（3）发信通过 browse_send_mail.php 和 browse_send_mail_content.php 两个文件来完成。通过 browse_send_mail.php 文件输出数据库中存储的发送记录，并且根据信息的标题进行分页输出，设置超链接，链接到 browse_send_mail_content.php 文件，在该文件中输出发送信息的详细内容。

由于篇幅所限，其他两个功能的程序代码这里没有给出，完整代码请参看本书附带光盘中的内容。

7.7.3　我的好友管理

我的好友功能也是从 7.4.3 节 send_forum_content.php 文件中设置的超链接开始。“我的好友”链接指向 my_friend.php 文件，在该文件中完成添加好友的操作，并且向好友发送一条验证信息。其运行结果如图 7.40 所示。

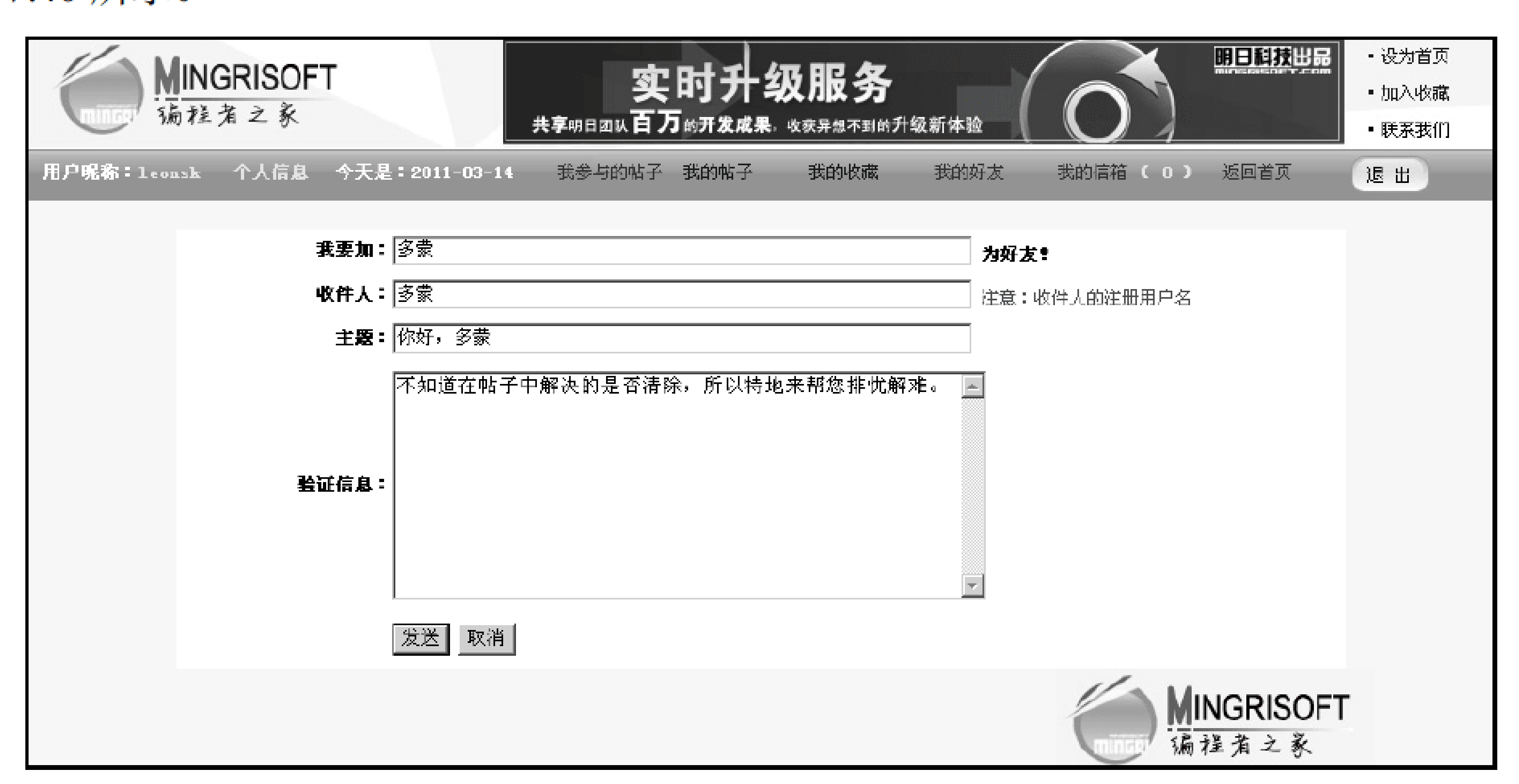
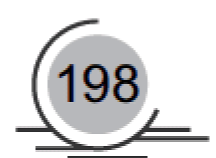


图 7.40　我的好友模块

在 my_friend.php 文件中，创建 form 表单，实现好友的提交，将数据提交到 my_friend_ok.php 文件中进行处理。在提交到处理页 my_friend_ok.php 后，首先需要判断申请的好友对象是否为自己，如果添加对象是自己，则给出警告提示。然后通过查询语句查询双方好友列表，判断申请者和被申请者是否至少有一方已存在对方名称，如果至少有一方存在对方名称，则同样不能执行添加操作，并返回警告提示；如果双方好友列表中都不存在对方会员名，则执行提交，并向被申请者发送一条验证信息。在得到对方验证之前，申请者不能重复进行好友添加操作，直到对方验证并同意添加好友，则完成添加好友操作过程。其中，my_friend_ok.php 文件的程序代码如下：

```
<?php session_start(); include("conn/conn.php");          //连接数据库
$tb_my=$_POST["my"];                                          //获取表单中提交的数据
```



```
$tb_friend=$_POST["friend"];                            //获取表单中提交的数据
$tb_date=date("Y-m-d");                                 //获取当前时间
$tb_receiving_person=$_POST["receiving_person"];        //获取接收人
$tb_mail_subject=$_POST["mail_subject"];                //获取信息主题
$tb_mail_content=$_POST["mail_content"];                //获取信息内容
$tb_mail_sender=$_POST["mail_sender"];                  //获取发送人
$tb_mail_date=date("Y-m-d");
$tb_friend_tmp=$tb_my;                                  //对接收人和发送人重新赋值
$tb_my_tmp=$tb_friend;
if($tb_my==$tb_friend){                                 //判断添加的对象是否为自己
    echo "<script>alert('你想自己加自己吗？');history.back();</script>";
}else{
    $querychkk=mysql_query("select tb_friend from tb_my_friend where tb_my='$tb_my_tmp'");
   $jcount=0;                                           //判断对方好友列表中是否存在自己名称
   while($myrow=mysql_fetch_array($querychkk)){
        if($myrow['tb_friend']==$tb_my)
            $jcount++;
        }
   $kcount=0;                                           //判断自己好友列表中是否存在对方名称
   $querychk=mysql_query("select tb_friend from tb_my_friend where tb_my='$tb_my'");
      while($myrow2=mysql_fetch_array($querychk)){
         if($myrow2['tb_friend']==$tb_friend)
             $kcount++;
         }
   if($kcount>0) {
      if($jcount>0)                                     //如果双方好友列表中存在对方名称
           echo "<script>alert('你们已经是好友了，不必再进行添加了');history.back();</script>";
     else if($jcount==0)                                //如果对方好友列表不存在自己名称
          echo "<script>alert('好友验证已发送，请耐心等待对方确认');history.back();</script>";
         }else{                         //如果双方好友列表中都不存在对方名称，则进行添加好友操作
      $querys=mysql_query("select * from tb_forum_user where tb_forum_user='$tb_receiving_person'");
        if(mysql_num_rows($querys)>0){
            $querys=mysql_query("insert into tb_my_friend(tb_my,tb_friend,tb_date)values('$tb_my','$tb_friend',
'$tb_date')");
            $query=mysql_query("insert into tb_mail_box(tb_receiving_person,tb_mail_subject,tb_mail_content,
tb_mail_sender,tb_mail_date,yanzheng )values('$tb_receiving_person','$tb_mail_subject','$tb_mail_content','$t
b_mail_sender','$tb_mail_date',1)");
                if($query==true){
                    echo "<script>alert('短信息发送成功!');history.back();</script>";
                }
        }else{
     echo "<script>alert('对不起，不存在该用户!');history.back();</script>";
        }
       }
}
?>
```

在完成好友的添加之后，在会员登录成功的页面中有一个“我的好友”超链接，单击该链接进入 browse_friend.php 文件中，在该文件中可以查看到所有的好友。当单击好友的名称时将链接到 person_

data.php 文件，通过该文件可以查看到好友的详细信息。在 browse_friend.php 文件中，还创建了一个 form 表单，将数据提交到 delete_friend.php 文件中，实现对指定好友进行删除的操作。删除过程同样可以参考 7.2.8 节中的内容，虽然整体思路相同，但是代码有细微变动。其运行结果如图 7.41 所示。



图 7.41　我的好友管理

7.8　后　台　管　理

7.8.1　后台管理概述

论坛后台管理作为整个论坛模块中不可缺少的因素之一，其不但可以实现各种帖子的删除、置顶、结帖等管理操作，而且可以对论坛会员的账号执行权限设置、账号删除等操作。其中，后台管理主要包括会员管理、回帖管理、顶帖管理等多个管理模块，其运行结果如图 7.42 所示。



图 7.42　后台管理功能展示

7.8.2　后台登录

和前台登录一样，管理员要想对论坛进行管理，首先必须执行登录操作，在管理员登录成功后，页面才能跳转到论坛管理后台主页。管理员与普通用户的身份判别是根据数据表 tb_forum_user 中的 tb_forum_type 字段来决定的，当该字段值为 1 时，用户身份为普通会员；当其值为 2 时，表明该用户身份为管理员。

后台管理员登录由 enter_manage.php 与 enter_manage_ok.php 两个文件组成。

（1）enter_manage.php 用于显示后台登录界面，该文件由两部分组成，第一部分为 JavaScript 内容，用于判断输入的管理员账号、密码以及验证码是否为空，其关键代码如下：

```
<script language="JavaScript" type="text/javascript">
 function check_user(form){
   if(form.tb_user.value==""){                           //判断用户名是否为空
        alert("请输入管理名");
      form.tb_user.select();
      return(false);
     }
if(form.tb_pass.value==""){                              //判断输入的密码是否为空
        alert("请输入登录密码！");
      form.tb_pass.select();
      return(false);
      }
     if(form.tb_validate.value==""){                     //判断输入的验证码是否为空
        alert("请输入验证码！");
      form.tb_validate.select();
      return(false);
      }
   return(true);
 }
</script>
```

第二部分为表单设计部分，用于获取管理员登录的用户名、密码、验证码等登录信息，其关键代码如下：

```
<table align="center" width="750" height="24" border="0" cellpadding="0" cellspacing="0" background="images/
bg_13(1).JPG">
<form action="enter_manage_ok.php" method="post" name="form1" id="form1" onSubmit="return check_user
(this)">
  <tr>
    <td width="71" align="right"><span class="STYLE1">用户名：</span></td>
    <td width="138"><input type="text" name="tb_user" size="18"   /></td>
    <td width="68" align="right" class="STYLE1">密码：</td>
    <td width="145"><input type="password" name="tb_pass" size="18" /></td>
    <td width="80" align="right" class="STYLE1">验证码：</td>
    <td width="94"><input type="text" name="tb_validate" size="10" /></td>
    <td width="50" align="center"><img src="../tb_validate.php"></td>
    <td width="87" align="center"><input type="submit" name="Submit" value="登录"></td>
    <td width="17"> </td>
  </tr>
</form>
</table>
```

（2）enter_manage_ok.php 封装一个 check_user 类，定义 check_input()方法对表单中提交的用户名和密码进行验证。关键代码如下：

```
<?php    session_start();
$tb_validate=trim($_POST[tb_validate]);
class check_user{
    var $tb_user;                                          //定义用户名变量
    var $tb_pass;                                          //定义用户密码变量
    var $tb_validate;                                      //定义验证码变量
    function check_user($x,$y,$m){                         //创建构造函数
     $this->tb_user=$x;
     $this->tb_pass=$y;
      $this->tb_validate=$m;
    }
    function check_input(){
      if(strval($this->tb_validate)!=$_SESSION["validate1"]){        //判断用户输入的验证码是否正确
          echo "<script>alert('验证码输入错误!');history.go(-1);</script>";
          exit;
       }
      include_once("conn/conn.php");                      //加载数据库相关配置
     $sql=mysql_query("select tb_forum_user from tb_forum_user where tb_forum_type=2 and tb_forum_
user='".$this->tb_user."'",$conn);                         //执行数据库查询
     $info=mysql_fetch_array($sql);
      if($info==false){                                    //判断用户名是否存在
          echo "<script>alert('对不起，不存在该用户!');history.back();</script>";
          exit;
     }else{
          $sql=mysql_query("select tb_forum_user from tb_forum_user where tb_forum_type=2 and tb_forum_
user='".$this->tb_user."' and tb_forum_pass='".$this->tb_pass."'",$conn);
          $info=mysql_fetch_array($sql);
          if($info==false){                                //判断输入的密码是否正确
              echo "<script>alert('对不起，密码输入错误!');history.back();</script>";
              exit;
           }else{
              if($_SESSION["admin_user"]!=""){             //判断 Session 变量是否为空
                  session_unregister("admin_user");        //销毁 Session 变量
               }
              $_SESSION["admin_user"]=$this->tb_user;      //重新注册 Session 变量
               echo "<script>alert('登录成功!');window.location.href='index.php';</script>";
           }
       }
      mysql_close($conn);                                  //关闭数据库连接
    }
  }
  $chk=new check_user($_POST[tb_user],md5($_POST[tb_pass]),$tb_validate);      //初始化该类
  $chk->check_input();                                                         //调用类的函数
?>
```

7.8.3 后台管理主页设计

后台管理主页的页面设计其实没有太多实质性的技术可言，其主要是为不同的管理栏目设置超链





接，然后应用 switch 语句根据超链接传递的参数值进行判断，进而完成不同栏目的加载功能。

在 index.php 文件中，首先判断当前访问管理页的用户权限。关键代码如下：

```
<?php
 session_start();                                              //开启 Session 设置
 include("../conn/conn.php");                                  //加载数据库配置文件
 if ($page=="") {                                              //判断当前页码是否为空
$page=1;
}
 if($_SESSION["admin_user"]==""){                              //判断当前用户权限
        echo "<script>alert('禁止非法登录!');window.location.href='enter_manage.php';</script>";
        exit;
 }else{
?>
```

然后为不同的栏目创建超链接，并且定义超链接的参数值。关键代码如下：

```
<table width="100%" height="275" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td height="80" colspan="3"><img src="images/index_2.jpg" width="1003" height="80"></td>
  </tr>
  <tr>
    <td height="24" colspan="3" ><img src="../images/index_4.jpg" width="1003"></td>
  </tr>
  <tr>
    <td width="153" valign="top" background="images/index_5.jpg"><table width="150" height="193" border=
"0" cellpadding="0" cellspacing="0" background="images/index_5.jpg">
      <tr>
        <td height="24" align="center"><a href="index.php?title=会员管理" class="STYLE3">会员管理</a>
</td>
      </tr>
      <tr>
        <td height="24" align="center"><a href="index.php?title=公告管理" class="STYLE3">公告管理</a>
</td>
      </tr>
      <tr>
        <td height="25" align="center"><a href="index.php?title=帖子类别管理" class="STYLE3">帖子类别管
理</a></td>
      </tr>
<!--省略部分代码-->
</table></td>
    <td width="7" height="400" bgcolor="#EFF3F7"> </td>
    <td width="1075" align="left" valign="top">
```

最后应用 switch()语句，根据 URL 地址中的 title 参数进行判断，加载不同的栏目页面，并将加载的内容输出到指定的层中，其关键代码如下：

```
<td width="1075" align="left" valign="top">
<div style=" width:840px;">
```

```
<?php
if(empty($_GET["title"]))                    //获取相关 URL 参数
$title="";
else
$title=$_GET["title"];
switch($title){                              //通过 GET 方法获取 URL 中参数 title 的值
	case "会员管理":
	include("leaguer_admin.php");            //加载相应的页面
	break;
	case "公告管理":
	include("send_affiche.php");
	break;
	case "帖子类别管理":
	include("append_small_type.php");
	break;
	//省略部分代码
case "":
	include("send_affiche.php");
	break;
}
?>
</div>
</td>
```

7.9 本章小结

本章运用软件工程的设计思想，通过一个完整的论坛来讲解一个系统的开发流程。同时，在论坛的开发过程中，讲解了如何屏蔽回帖，以及帖子的发布、回复、收藏等功能，使整个系统的设计思路更加清晰。通过本章的学习，读者不仅可以了解一般网站的开发流程，而且可以对论坛中的内容进行回帖、发表帖子等。



第 8 章

在线支付模块

（PHP+Smarty+MySQL 实现）

随着 Internet 的发展，电子商务已经成为 21 世纪网络发展的主流，网上购物和网上支付已经成为一种时尚。目前，国内企业和个人都纷纷加入到阿里巴巴等一系列大型电子商务网站中，通过网络进行 B2B、B2C 和 C2C 的网上交易，在进行网上交易的同时，付款方法也在随着网络的发展发生着变化，在线支付方式的快捷和方便已经得到越来越多人的认可。为了让广大 PHP 爱好者能开发出功能更加完善的电子商务类网站，在本模块中，将介绍如何通过支付宝完成电子商务网站在线支付操作，以及订单生成和订单打印技术的实现方法。

通过阅读本章，可以学习到：

- 动态生成订单号
- 订单打印
- 支付宝支付
- 订单查询
- 统计购物车中商品金额
- 向购物车中添加商品
- 清空购物车
- Smarty 模板实现网页动静分离

8.1 在线支付模块概述

8.1.1 功能概述

在线支付与购物车相同，都是电子商务网站必不可少的一部分。本模块在购物车模块的基础上，将填写收货人信息、生成订单、订单打印、订单预览和订单查询作为辅助，最终完成支付宝在线支付的操作。在线支付模块的功能结构如图 8.1 所示。

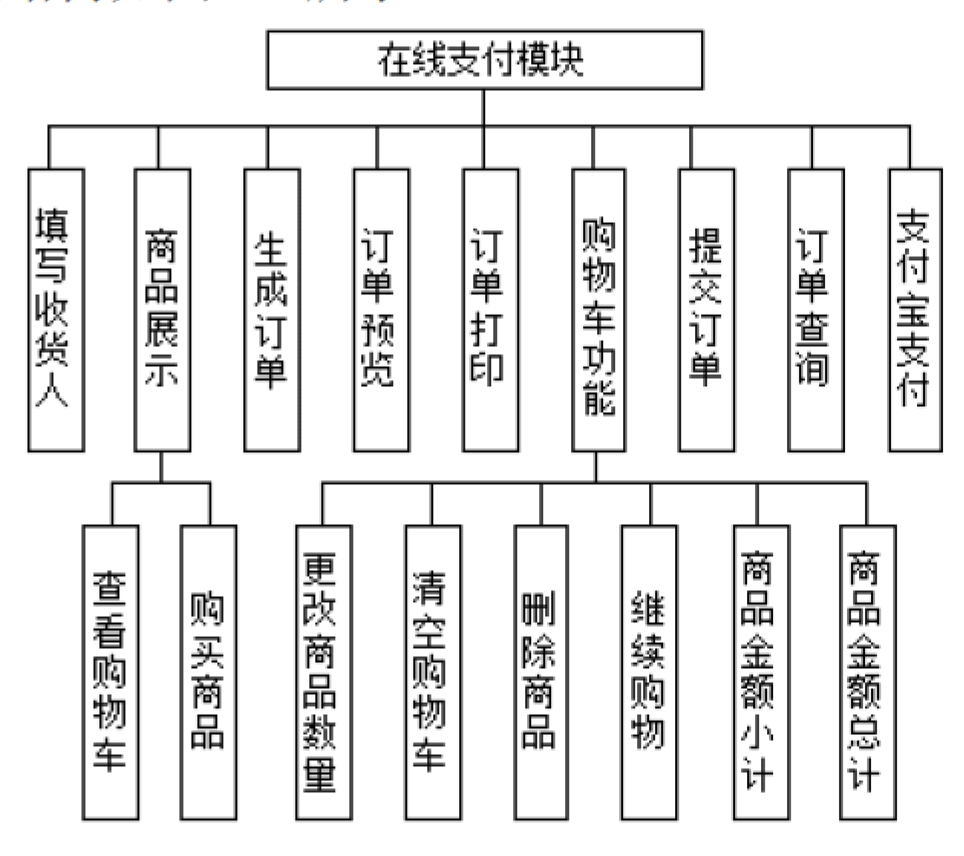


图 8.1　在线支付模块的功能结构图

8.1.2 在线支付操作流程

在线支付模块的操作流程与购物车十分相似，唯一区别就是增加了支付宝支付和订单查询的功能。其流程如图 8.2 所示。

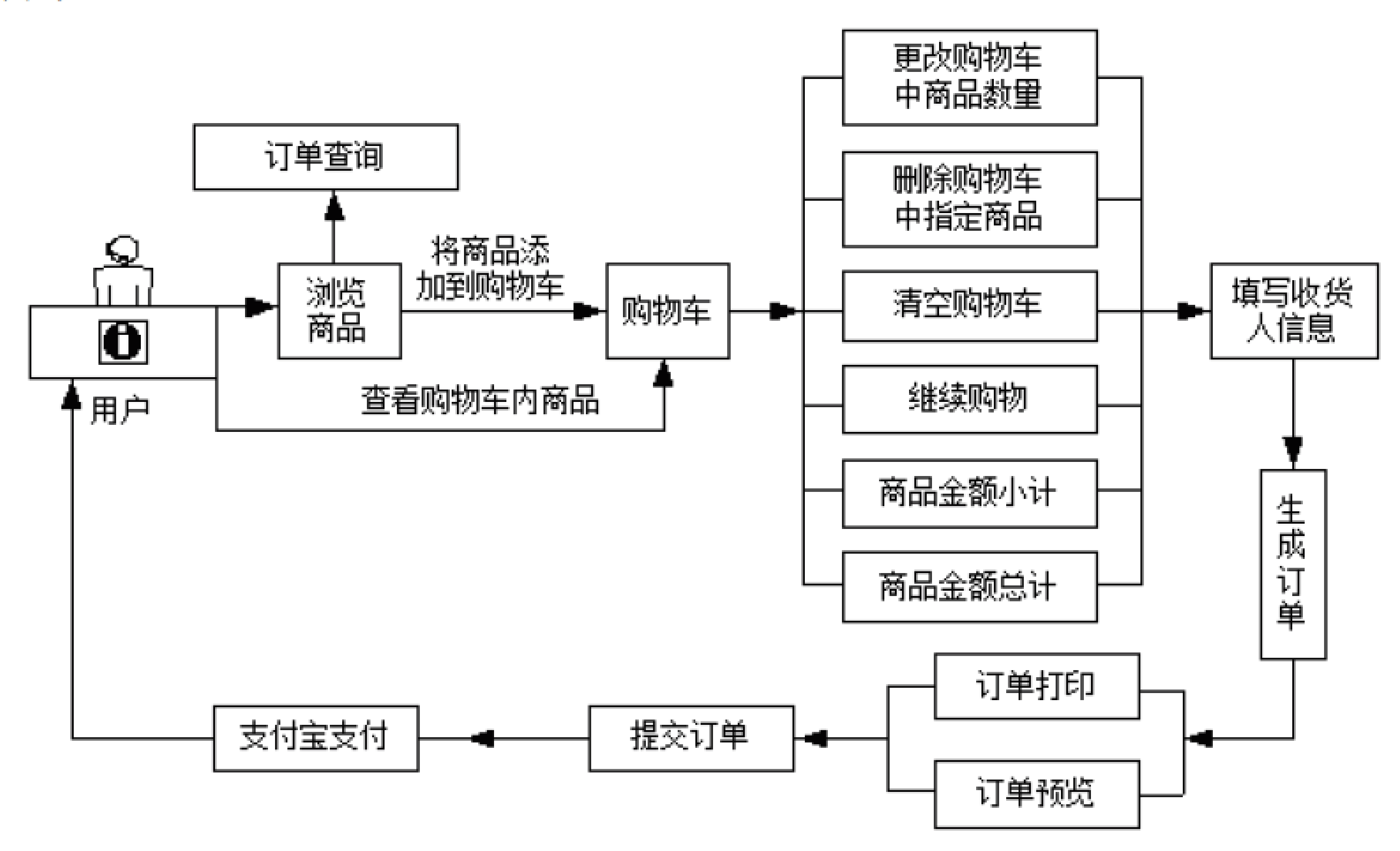


图 8.2　在线支付的操作流程





8.1.3　程序预览

在线支付模块是电子商务网站收银台中的一个子功能，为了让读者对本模块有个初步的了解和认识，下面列出几个典型功能的页面。

明日购物商城网站填写收货人信息的页面如图 8.3 所示，该页面完成收货人姓名、电话、联系地址等信息的添加操作。

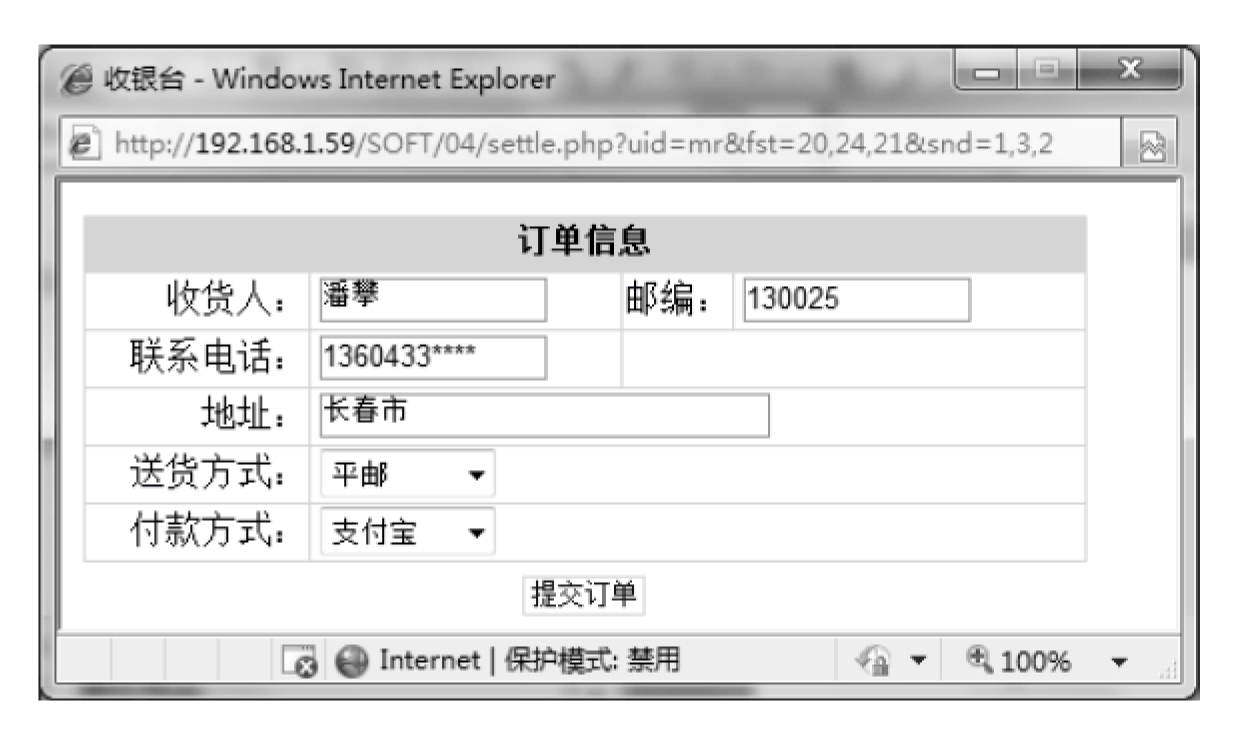


图 8.3　收货人信息添加页面

收货人信息填写完毕之后，就是提交订单，并且生成订单号，进而完成订单的预览、打印以及在线支付等操作。生成订单信息的页面如图 8.4 所示。



图 8.4　生成订单信息页面

在图 8.4 中单击“打印预览”按钮，将进入打印预览页面，其运行效果如图 8.5 所示。

单击“我要打印”按钮，将执行订单的打印操作，其运行效果如图 8.6 所示。



图 8.5　订单打印预览页面

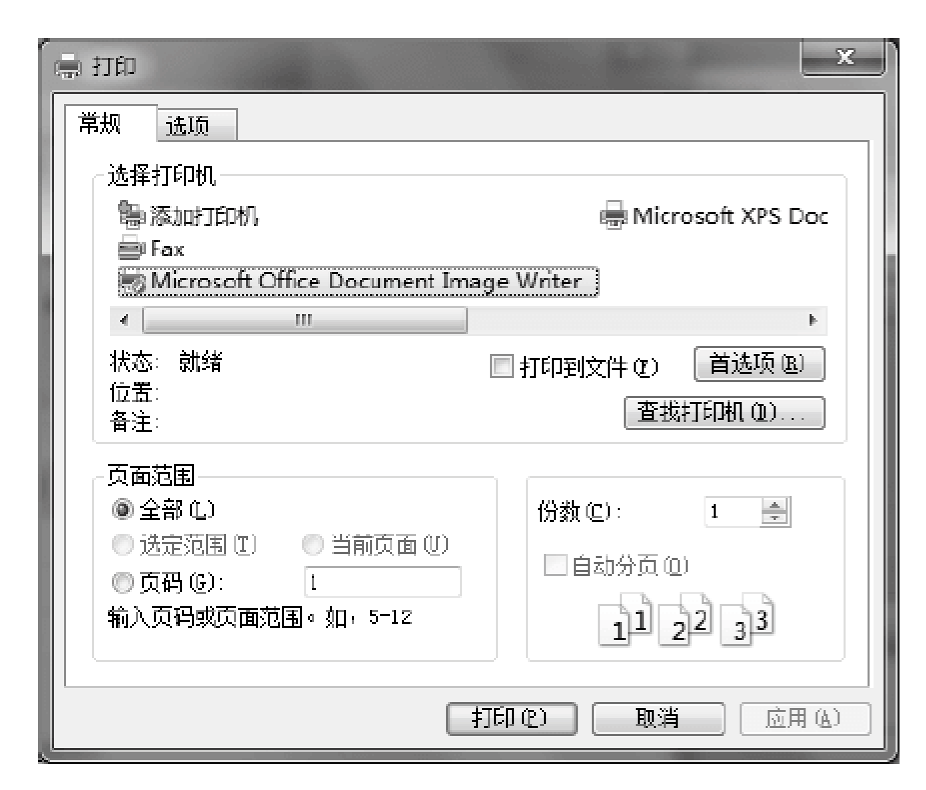


图 8.6　订单打印

8.2　数据库设计

本模块与购物车模块同属于电子商务网站中一个子功能，其数据库的使用是统一的，都是 MySQL 数据库。与购物车模块存在区别的地方就是调用的是不同的数据表。

8.2.1　创建数据库

再一次看一下电子商务网站所使用的数据表情况，如图 8.7 所示。


服务器: localhost ▸ 数据库: db_business

结构　SQL　搜索　查询　导出　Import　操作　权限　删除

表	操作	记录数	类型	整理	说明
tb_admin		2	MyISAM	utf8_unicode_ci	管理员表
tb_class		13	MyISAM	utf8_unicode_ci	商品类别表
tb_commo		4	MyISAM	utf8_unicode_ci	商品信息表
tb_form		17	MyISAM	utf8_unicode_ci	商品订单表
tb_links		2	MyISAM	utf8_unicode_ci	友情链接表
tb_opinion		0	MyISAM	utf8_unicode_ci	商品留言表
tb_public		4	MyISAM	utf8_unicode_ci	商品公告表
tb_user		6	MyISAM	utf8_unicode_ci	会员信息表
8 个表	总计	48	MyISAM	utf8_general_ci	

图 8.7　电子商务网站数据表



8.2.2　创建数据表

本节介绍与在线支付有直接关系的数据表的结构。

☑　tb_commo（商品信息表）

商品信息表主要用于存储关于商品的相关信息，其结构如图 8.8 所示。

服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_commo

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	utf8_unicode_ci		否			商品名称
pics	varchar(200)	utf8_unicode_ci		否	pics/null.jpg		商品图片
info	mediumtext	utf8_unicode_ci		否			商品介绍
addtime	date			否			添加时间
area	varchar(50)	utf8_unicode_ci		否			商品产地
model	varchar(50)	utf8_unicode_ci		否			商品型号
class	varchar(50)	utf8_unicode_ci		否			商品类型
brand	varchar(50)	utf8_unicode_ci		否			商品名称
stocks	int(4)			否	1		商品库存
sell	int(4)			否	0		销售量
m_price	float			否			市场价格
v_price	float			否			会员价格
fold	float			否	9		打折率
isnew	int(1)			否	1		是否新品
isnom	int(1)			否	0		是否推荐

图 8.8　商品信息表结构

☑　tb_form（订单信息表）

订单信息表主要用于存储关于订单的相关信息，其结构如图 8.9 所示。

服务器: localhost ▸ 数据库: db_business ▸ 表 : tb_form

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
formid	varchar(125)	utf8_unicode_ci		否			订单号
commo_id	varchar(100)	utf8_unicode_ci		否			商品ID
commo_name	varchar(50)	utf8_unicode_ci		否			商品名称
commo_num	varchar(100)	utf8_unicode_ci		否			商品数量
agoprice	varchar(50)	utf8_unicode_ci		否			商品单价
fold	varchar(50)	utf8_unicode_ci		否			打折率
total	varchar(50)	utf8_unicode_ci		否			消费总额
vendee	varchar(50)	utf8_unicode_ci		否			消费者
taker	varchar(50)	utf8_unicode_ci		否			收货人姓名
address	varchar(200)	utf8_unicode_ci		否			收货地址
tel	varchar(20)	utf8_unicode_ci		否			联系电话
code	varchar(10)	utf8_unicode_ci		否			邮编
pay_method	varchar(20)	utf8_unicode_ci		否			付款方式
del_method	varchar(20)	utf8_unicode_ci		否			邮寄方式
formtime	timestamp			否	CURRENT_TIMESTAMP		订单日期
state	int(1)			否			订单状态

图 8.9　订单信息表结构

此外，还有管理员表、商品公告表、用户信息表、友情链接表和商品留言表，限于篇幅，这里不再介绍，读者可参见本书附赠光盘中的数据库文件。



8.3 会员信息模块设计

8.3.1 会员信息模块概述

用户登录后，即可看到会员信息模块。在这里，可以进行查看或修改个人信息及密码、查看购物车和安全退出等操作。本节只对会员信息模块中的“会员中心”和“安全退出”进行讲解。会员信息模块的运行效果如图 8.10 所示。



图 8.10 会员信息模块的运行效果

8.3.2 会员中心

当单击“会员中心”超链接时，会回传给当前页一个 page 值，当前页根据这个 page 值来载入 member.php 文件。

1. 创建 PHP 页面

与登录模块设计不同，本节首先来创建 PHP 页面。因为该模块中的模板需要使用数据库中的数据及一些动态信息，这些都需要在 PHP 页中先行获取及处理，然后再传给模板页。会员中心页面的代码请参考 8.6 节技术提炼分析中的内容。

2. 创建模板页

该模块包括查看信息模板及修改密码模板，都存储于 member.tpl 模板文件中。关键代码如下：

```
<link rel="stylesheet" href="css/member.css" />
<script language="javascript" src="js/member.js"></script>
{if $check=="find" }
<p align="left">{$smarty.session.member}&gt;&gt;&gt;<a href='?page=hyzx' id="mem">查看信息</a>&gt;&gt;
<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
<table   id="member" width="300" border="0" cellpadding="0" cellspacing="0">
<form  id="member"  name="member"  method="post"  action="modify_pwd_chk.php"  onSubmit="return  pwd
(member)">
   <tr>
<td height="25" colspan="2" align="center" valign="middle" id="first"><font color="#f0f0f0">修改密码</font></td>
</tr>
…//省略了部分代码
<tr>
<td height="30" colspan="2" align="center" valign="middle">
<input id="enter" name="enter" type="submit" value="修改" /></td>
</tr>
```





```
</form>
</table>
{else}
<p align="left">{$smarty.session.member}&gt;&gt;&gt;<a href='?page=hyzx' id="mem">查看信息</a>&gt;
<a href='?page=hyzx&action=modify' id="mem">修改密码</a></p>
{section name=pwd_id loop=$pwdarr}
<table id='member' width="500" border="0" cellpadding="0" cellspacing="0">
<form id="member" name="member" method="post" action="modify_info_chk.php" onSubmit="return mem
(member)">
<tr>
<td height="25" colspan="2" align="center" valign="middle" id="first">
<font color="#f0f0f0">{$pwdarr[pwd_id].name}信息（不可更改信息）</font></td>
</tr>
<tr>
  <td width="25%" height="25" align="right" valign="middle" id="left"> 会员编号：</td>
<td height="25" align="left" valign="middle" id="right"> {$pwdarr[pwd_id].id}</td>
</tr>
…//省略了部分代码
<tr>
<td height="30" colspan="2" align="center" valign="middle">
<input name="enter" type="submit" id="enter" value="修改" />    
<input name="reset" type="reset" id="reset" value="重置" /></td>
  </tr>
</form>
</table>
{/section}
{/if}
```

3．创建脚本文件

该模块的脚本文件和用户注册模块类似，都是对信息的合法性进行验证，如信息是否为空、是否符合规范等，其存储于 08\js\member.js 中，这里不再赘述。

4．创建处理页

当信息验证通过后，系统将跳转到处理页 modify_info_chk.php 进行信息处理。信息处理页代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );        //设置文件编码格式
require("system/system.inc.php");                           //包含配置文件
$sql = "update tb_user set realname='".$_POST['realname']."',card='".$_POST['card']."',tel='".$_POST['tel']."',
phone='".$_POST['phone']."',Email='".$_POST['email']."',QQ='".$_POST['qq']."',code='".$_POST['code']."',
address='".$_POST['address']."' where id = '".$_POST['userid']."'";
$arr = $admindb->ExecSQL($sql,$conn);
if($arr)
echo "<script>alert('修改成功');location=('index.php');</script>";
else
echo "<script>alert('修改失败');history.go(-1);</script>";
?>
```

8.3.3 安全退出

当用户需要离开网站时，可以单击“安全退出”超链接来调用 logout()函数，当用户确认退出后，则跳转到 logout 页面，销毁 Session 并回到首页。安全退出所涉及的页面及代码如下：

代码位置：光盘\mr\08\js\info.js

```
function logout(){
if(confirm("确定要退出登录吗？ ")){                    //输出选择框，用户可以单击“确认”或“取消”按钮
      window.open('logout.php','_parent','',false);    //如果用户确认退出，则打开 logout.php 页
}else
   return false;
}
```

代码位置：光盘\mr\08\logout.php

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );      //设置文件编码格式
session_destroy();
echo '<script>alert(\'用户已安全退出!\');location=(\'index.php\');</script>';
?>
```

8.4 收银台模块设计

8.4.1 收银台模块概述

当用户停止浏览商品准备结账时，可以单击购物车页面中的“去收银台”按钮，将触发 onclick 事件，调用 formset()函数显示订单页面，当用户提交订单后，系统将订单保存到数据表 tb_form 中，同时清空购物车，并显示订单信息，提醒用户记录订单号。当货款发出后，还可以对订单进行查询。收银台页面的运行结果如图 8.11 所示。

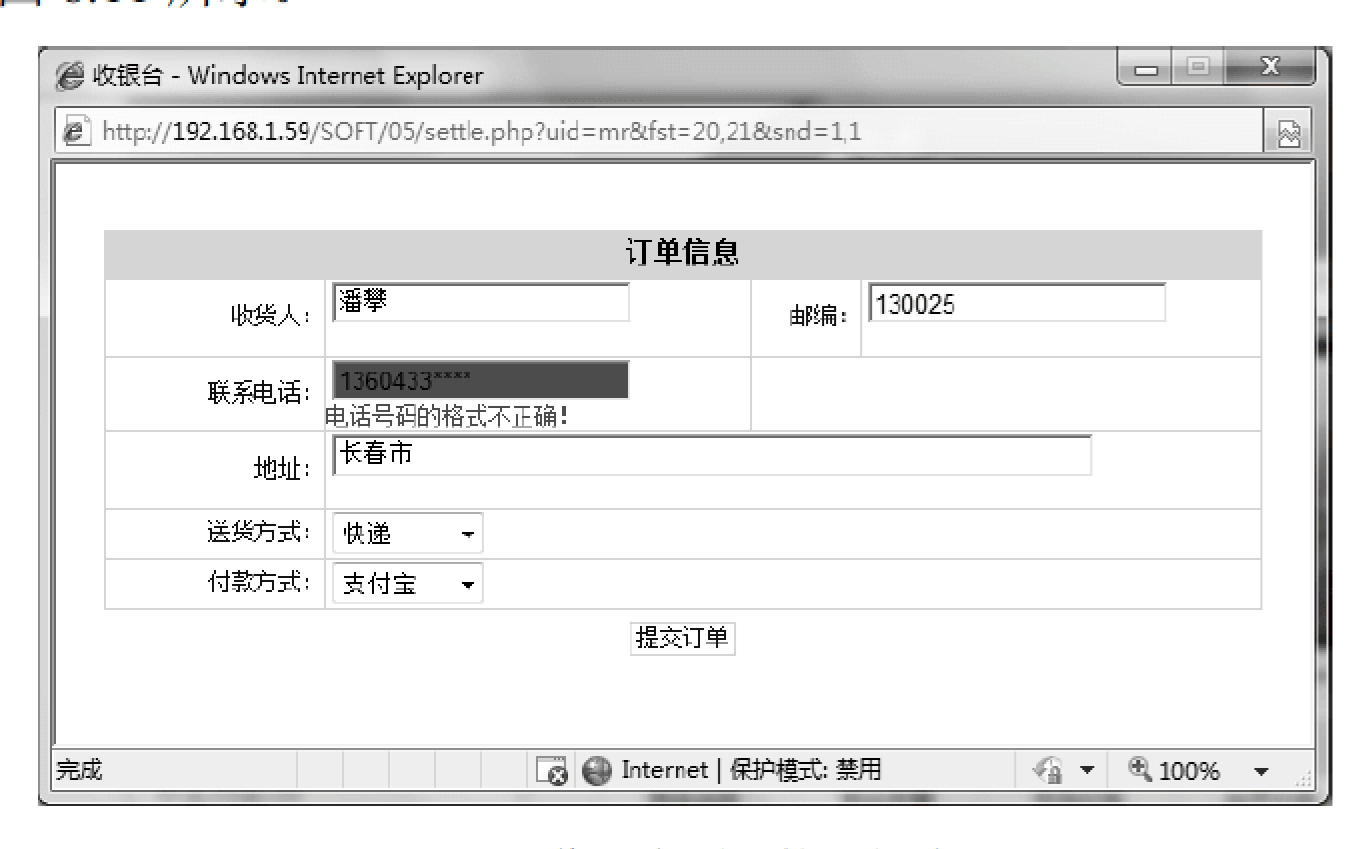


图 8.11 收银台页面的运行结果



本节所涉及的页面有显示订单（formset()函数）、填写订单（settle.php、settle.tpl）、提交订单（settle_chk.php）、反馈订单（forminfo.php、forminfo.tpl）和查询订单 5 部分。

8.4.2　显示订单

订单信息提交页面的输出由 formset()函数决定，它将商品信息进行整理，通过 open()方法打开 settle.php 页来显示订单，并将整理后的商品信息传递到 settle.php 文件中。formset()函数存储于 js\shopcar.js 文件中，其关键代码如下：

```
function formset(form){
var uid = form.uid.value;
var n_pre = 'cnum';                                    //数量
var lang = form.chk.length;
if(lang == undefined){
var fst = form.chk.value;                              //商品 id
var snd = form.cnum0.value;                            //购买数量
}else{
var fst= new Array();
var snd = new Array();
for(var i = 0; i < lang; i++){
var nm = n_pre+i.toString();
var stmp = document.getElementById(nm).value;
if(stmp   == '' || isNaN(stmp)){
alert('不允许为空，必须为数字');
document.getElementById(nm).select();
return false;
}
snd[i] = stmp;
var ftmp = form.chk[i].value;
fst[i] = ftmp;
}
}
open('settle.php?uid='+uid+'&fst='+fst+'&snd='+snd,'_blank','width=500 height=450',false);
}
```

说明　因为 open()方法使用了_blank 参数来打开一个新的页面，Session 值传不过去，所以这里使用隐藏域来传递用户名称。

8.4.3　填写订单

settle.php 直接将接收的值传给 settle.tpl 模板，并载入 settle.tpl 模板。settle.php 页面的代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
```

```
require("system/system.inc.php");                                    //包含配置文件
$fst = $_GET['fst'];
$snd = $_GET['snd'];
$uid = $_GET['uid'];
$smarty->assign('title','收银台');
$smarty->assign('fst',$fst);
$smarty->assign('snd',$snd);
$smarty->assign('uid',$uid);
$smarty->display('settle.tpl');
?>
```

在 settle.tpl 模板页首先载入 settle.js 脚本文件，调用其中的 chkreginfo()方法验证表单提交的值是否符合要求。然后创建表单 form_reg，提交收货人、联系电话、联系地址等信息，同时还将 PHP 页传递的变量以隐藏域的形式提交到处理页 settle_chk.php。settle.tpl 模板页的关键代码如下：

```
<form name="form_reg" method="post" action="settle_chk.php" onSubmit="return chkreginfo(form_reg,'all')">
<tr>
<td height="25" align="right" valign="middle" class="left">收货人：</td>
<td width="221" height="25" align="left" valign="middle" class="center"> 
<input type="text" name="taker" size="20" onBlur="chkreginfo(form_reg,0)">
<div id="chknew_taker" style="color:#FF0000"></div>
</td>
<td width="56" align="right" valign="middle" class="center">邮编：</td>
<td width="207" align="left" valign="middle" class="right"> 
<input type="text" name="code" size="20" onBlur="chkreginfo(form_reg,1)">
<div id="chknew_code" style="color:#FF0000"></div></td>
</tr>
<!--省略了部分代码-->
<tr>
<td width="116" height="25" align="right" valign="middle" class="left"> 付款方式：</td>
<td height="25" colspan="3" align="left" valign="middle" class="right"> <select id="pay" name="pay">
<option value="银行转账">银行转账</option>
<option value="邮局汇款">邮局汇款</option>
<option value="支付宝">支付宝</option>
</select>      </td>
</tr>
<tr>
<td height="30" colspan="4" align="center" valign="middle">
<input id="enter" name="enter" type="submit" value="提交订单" class="btn" />
<input id="fst" name="fst" type="hidden" value="{$fst}" />
<input id="snd" name="snd" type="hidden" value="{$snd}" />
<input id="uid" name="uid" type="hidden" value="{$uid}" ></td>
</tr>
</form>
```

8.4.4 处理订单

处理页 settle_chk.php 获取表单中提交的数据，根据用户提交的商品信息，重新查找数据表 tb_commo，



并从数据表中提取商品信息，保存到数组中，然后处理页将数组作为一条记录添加到 tb_form 表。

数据添加成功的同时，处理页会根据 uid 找到该用户，将 shopping 字段清空，最后调用 forminfo.php 页来显示新添加的订单信息。settle_chk.php 页的代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
require("system/system.inc.php");                                 //包含配置文件
$sql="insert into tb_form(formid,commo_id,commo_name,commo_num,agoprice,fold,total,vendee,taker,address,
tel,code,pay_method,del_method,formtime,state)values(";
$formid=time();
$tmpid = explode(',',$_POST['fst']);
$tmpnm = explode(',',$_POST['snd']);
$number = count($tmpid);
$tmpna = array();
$tmpvp = array();
$tmpfd = array();
$tmptt = 0;
if($number >1){
for($i = 0; $i < $number; $i++){
$tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[$i]."'";
$tmprst = $admindb->ExecSQL($tmpsql,$conn);
$tmpna[$i] = $tmprst[0]['name'];
$tmpvp[$i] = $tmprst[0]['v_price'];
$tmpfd[$i] = $tmprst[0]['fold'];
$tmptt += $tmprst[0]['v_price'] * $tmpnm[$i];
@$tmpsell = $tmprst[0]['sell'] + 1;
$addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[$i]."'";
$addrst = $admindb->ExecSQL($addsql,$conn);
}
$sql.="'".$formid."','".$_POST['fst']."','".implode(',',$tmpna)."','".$_POST['snd']."','".implode(',',$tmpvp)."',
'".implode(',',$tmpfd)."','".$tmptt."','".$_POST['uid']."'";
}else if($number == 1){
$tmpsql = "select name,v_price,fold from tb_commo where id = '".$tmpid[0]."'";
$tmprst = $admindb->ExecSQL($tmpsql,$conn);
$tmptt= $tmprst[0]['v_price'] * $tmpnm[0];
@$tmpsell = $tmprst[0]['sell'] + 1;
$addsql = "update tb_commo set sell = '".$tmpsell."' where id = '".$tmpid[0]."'";
$addrst = $admindb->ExecSQL($addsql,$conn);
$sql.="'".$formid."','".$_POST['fst']."','".$tmprst[0]['name']."','".$_POST['snd']."','".$tmprst[0]['v_price']."',
'".$tmprst[0]['fold']."','".$tmptt."','".$_POST['uid']."'";
}else{
echo 'error';
exit();
}
$sql.=",'".$_POST['taker']."','".$_POST['address']."','".$_POST['tel']."','".$_POST['code']."','".$_POST['pay']."',
'".$_POST['del']."','".date("Y-m-d H:i:s")."',0)";
$InsertSQL = $admindb->ExecSQL($sql,$conn);
if(false == $InsertSQL){
echo "<script>alert('购买失败');history.back;</script>";
}else{
```

```
$updsql = "update tb_user set consume='".$tmptt."',shopping='' where name = '".$_POST['uid']."'";
$updrst = $admindb->ExecSQL($updsql,$conn);
echo "<script>top.opener.location.reload();</script>";
echo "<script>open('forminfo.php?fid=$formid','_blank','width=750 height=650',false);window.close();</script>";
}
?>
```

8.4.5 生成订单

订单处理成功之后，将跳转到 forminfo.php 页输出生成的订单。订单生成的关键是 forminfo.php 和 forminfo.tpl。

在 forminfo.php 文件中，根据超链接传递的订单号，查询订单表中的数据，并且将查询结果赋给指定的模板变量，最终指定模板页 forminfo.tpl。关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include_once("system/system.inc.php");                              //包含类的实例化文件
$ddnumber=$_GET['fid'];                                             //获取订单号
$sql = "select * from tb_form where formid='".$_GET['fid']."'";    //定义 SQL 语句
$formarr = $admindb->ExecSQL($sql,$conn);                           //执行 SQL 语句
$commname = explode(',',$formarr[0]['commo_name']);                 //读取数据库中数据
$commnumber = explode(',',$formarr[0]['commo_num']);
$commagoprice = explode(',',$formarr[0]['agoprice']);               //读取数据库中数据
$commfold = explode(',',$formarr[0]['fold']);
$smarty->assign('formarr',$formarr[0]);                             //将数据赋给模板变量
$smarty->assign('commname',$commname);
$smarty->assign('commnumber',$commnumber);
$smarty->assign('commagoprice',$commagoprice);
$smarty->assign('commfold',$commfold);
if($formarr[0]["del_method"]=="平邮"){
$yprice="200";
}else{
$yprice="300";
}
…//省略部分代码
$smarty->assign('title','订单提交成功');
$smarty->display('forminfo.tpl');                                   //指定模板页
?>
```

在 forminfo.tpl 模板页中，获取模板变量传递的值，完成订单信息的输出，并且创建订单打印、打印预览和支付宝支付的按钮。关键代码如下：

```
<table width="650" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="30" colspan="5" align="center" valign="middle" class="first">订单内容</td>
</tr>
<tr>
<td width="100" height="25" align="center" valign="middle" class="left">商品名</td>
```



```
<td width="100" height="25" align="center" valign="middle" class="center">数量</td>
<td width="100" height="25" align="center" valign="middle" class="center">价格</td>
<td width="100" height="25" align="center" valign="middle" class="center">价格折率</td>
<td width="100" height="25" align="center" valign="middle" class="right">合计</td>
</tr>
{foreach key=key item=item from=$commname}
<tr>
<td height="25" align="center" valign="middle" class="left">{$item}</td>
<td height="25" align="center" valign="middle" class="center">{$commnumber[$key]}</td>
<td height="25" align="center" valign="middle" class="center">{$commagoprice[$key]} 元</td>
<td align="center" valign="middle" class="center">{$commfold[$key]} 折</td>
<td align="center" valign="middle" class="right">{$commagoprice[$key]*$commnumber[$key]} 元</td>
</tr>
{/foreach}
<tr>
<td colspan="5" height="25" align="right" valign="middle">总消费：{$formarr.total} 元</td>
</tr>
</table>
<input type="button" value="打印预览" onclick="printview()" class="btn" />  
<input type="button" value="我要打印" onclick="prints()" class="btn" />   
<!--<input type="button" value="我要打印" onclick="window.print()" class="btn" />-->
<!--支付宝支付的接口操作，提交的数据-->
  <a href="{$link}">支付宝支付</a>
<!-------------------------------------------------->
```

说明 本模块中的订单打印、打印预览和支付宝支付功能都是在 forminfo.php 和 forminfo.tpl 文件中完成的，有关这些内容将在 8.4.6 节中进行详细介绍。

8.4.6　订单预览、打印

在线支付模块中应用 WebBrowser 技术实现订单的打印和预览功能。打印和预览的运行效果如图 8.12 和图 8.13 所示。



图 8.12　订单打印



图 8.13　打印预览

本模块中的打印和预览功能在 forminfo.tpl 模板页中实现。首先载入 print.js 脚本文件；然后定义 object 标签；最后创建打印预览和打印的提交按钮，并且通过 onclick 事件调用对应的方法完成对应的操作。关键代码如下：

```
<script language="javascript" src="js/print.js"></script>
<object ID='WebBrowser1'   CLASSID='CLSID:8856F961-340A-11D0-A96B-00C04FD705A2'></object>
<input type="button" value="打印预览" onclick="printview()" class="btn" />
<input type="button" value="我要打印" onclick="prints()" class="btn" />   <!--
<input type="button" value="我要打印" onclick="window.print()" class="btn" />-->
```

print.js 脚本文件中封装了 printview()和 prints()两个方法，分别调用 WebBrowser 控件中的方法完成打印预览和打印的操作，其代码如下：

```
function printview(){
document.all.WebBrowser1.ExecWB(7,1);            //打印预览
window.close();
}
function prints(){
document.all.WebBrowser1.Execwb(6,1);            //打印
window.close();
}
```



8.5　订 单 查 询

8.5.1　订单查询功能概述

订单查询功能根据提交的订单号，从数据库中查询出指定订单的数据。其运行结果如图 8.14 所示。



图 8.14　订单查询

8.5.2　订单查询功能的实现

订单查询功能由 4 个文件组成，分别是 queryform.php、queryform.tpl、queryform.js 和 query.php。

（1）queryform.php 文件为动态 PHP 文件，用于指定模板页 queryform.tpl。

（2）queryform.tpl 模板页用于创建订单查询的表单。载入 JavaScript 脚本，通过 JavaScript 脚本中的 queryrst()方法调用 query.php 完成订单的无刷新查询，将查询结果在 queryform.tpl 模板页中输出。queryform.tpl 模板页的关键代码如下：

```
<form id='queryform' name='queryform' method='post' action='#'>
<tr>
<td height='25' colspan='5' align='center' valign='middle' class='first'>查询订单</td>
</tr>
<tr>
<td width='80' height='25' align='right' valign='middle' class='left'>查询用户：</td>
<td width='130' height='25' align='left' valign='middle' class='center'> 
<input id='name' name='name' type='text' class='txt' /></td>
```

```
<td width='100' height='25' align='right' valign='middle' class='center'>查询订单号：</td>
<td width='130' height='25' align='left' valign='middle' class='right'> 
<input id='formid' name='formid' type='text' class='txt' /></td>
<td width='100' align='center' valign='middle' class='center'>
<input id='enter' name='enter' type='button' value='查询' class='btn' onclick='return queryrst(queryform)'/></td>
</tr>
<tr>
<td height='25' colspan='5' align='center' valign='middle'> </td>
</tr>
</form>
```

（3）创建 queryform.js 脚本文件，定义 queryrst()和 showfm()方法，调用 query.php 文件完成订单的无刷新查询，并且将查询结果赋给 exam。代码如下：

```
// JavaScript Document
function queryrst(form) {
var name = form.name.value;                              //获取表单提交的值
var formid = form.formid.value;                          //获取表单提交的值
if(name == '' && formid == ''){                          //判断提交的值是否为空
alert('用户或订单号至少有一个不能为空');
form.name.focus();
return false;
}
var url = "query.php?vendee="+name+"&formid="+formid;   //定义调用的 URL
xmlhttp.open("GET",url,true);                            //执行调用操作
xmlhttp.onreadystatechange = showfm;                     //回调函数
xmlhttp.send(null);                                      //执行
}
function showfm(){
if(xmlhttp.readyState == 4){                             //判断返回结果
var msg = xmlhttp.responseText;
if(msg == '0'){
exam.innerHTML = '';
alert('没有查询结果');
}else{
exam.innerHTML = msg;                                    //将查询结果赋给 exam
}
}
}
```

（4）创建 query.php，根据超链接传递的参数值执行订单查询操作，并且返回查询结果。代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );     //设置文件编码格式
include_once("system/system.inc.php");                   //载入类的实例化文件，加载对象
$vendee = $_GET['vendee'];                               //获取超链接传递的值
$formid = $_GET['formid'];                               //获取订单号
$reback = '0';
$tmp = "<table width='600' border='0' align='center' cellpadding='0' cellspacing='0'>
<tr><td height='25' colspan='7' align='center' valign='middle' class='first'>查询结果</td></tr>
```



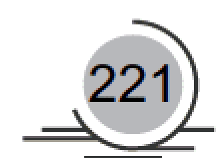


```
<tr><td width='100' height='25' align='center' valign='middle' class='left'>订单号</td>
<td width='75' height='25' align='center' valign='middle' class='center'>订货会员</td>
<td width='75' height='25' align='center' valign='middle' class='center'>收货人</td>
<td width='100' height='25' align='center' valign='middle' class='center'>订单金额</td>
<td width='75' height='25' align='center' valign='middle' class='center'>付款方式</td>
<td width='75' height='25' align='center' valign='middle' class='center'>收款方式</td>
<td width='100' height='25' align='center' valign='middle' class='right'>订单状态</td></tr>";
$sql = "select id,formid,vendee,taker,total,pay_method,del_method,state from tb_form
where vendee = '".$vendee."' or formid = '".$formid."'";
$rst = $admindb->ExecSQL($sql,$conn);                          //执行查询语句
if($rst){
foreach($rst as $value){                                        //循环输出查询结果
$tmp .= "<tr><td height=25 align=center valign=middle class=left>".$value['formid']."</td>
<td align=center valign=middle class=center>".$value['vendee']."</td>
<td align=center valign=middle class=center>".$value['taker']."</td>
<td   align=center valign=middle class=center>".$value['total']."</td>
<td   align=center valign=middle class=center>".$value['pay_method']."</td>
<td align=center valign=middle class=center>".$value['del_method']."</td>
<td align=center valign=middle class=right>";
switch ($value['state']){
case 0:
$tmp .= '未作处理';
break;
case 1:
$tmp .= '已付款';
break;
case 2:
$tmp .= '已发货';
break;
case 3:
$tmp .= '已收货';
break;
}
$tmp .= "</td></tr>";
}
$tmp .= "</table>";
$reback = $tmp;
}
echo $reback;
?>
```

8.6 技术提炼

8.6.1 收货人信息验证

为了保证网上交易的顺利完成，在处理收货人提交的信息时，必须对其填写的信息进行有效的验证，保证其填写的信息是合理有效的，避免格式上的错误出现。这里在验证收货人提交的信息时应用



的是 JavaScript 脚本和正则表达式。验证收货人信息的运行结果如图 8.15 所示。



图 8.15　验证收货人信息的运行结果

（1）验证手机号码格式

验证电话号码主要是通过 JavaScript 脚本的自定义函数调用正则表达式，实现对手机号码的验证。编写的判断手机号码格式的正则表达式如下：

```
/^13(\d{9})$|^15(\d{9})$/;
```

其判断手机号码是否是以 13 或者 15 开始，并判断是否是 11 位数字。判断手机号码格式是否正确的自定义函数的代码如下：

```
function checkregtel(regtel){
var str=regtel;
var Expression=/^13(\d{9})$|^15(\d{9})$/;  //判断手机号码的号段和格式是否正确
var objExp=new RegExp(Expression);
if(objExp.test(str)==true){
return true;
}else{
return false;
}
}
```

（2）验证收货人的其他信息

验证收货人的其他信息主要应用的是 JavaScript 脚本，通过 JavaScript 脚本中的自定义函数获取表单提交的数据，在自定义函数中判断数据是否合理，如果不合理，则通过 div 标签直接输出错误提示。验证收货人的其他信息的代码如下：

```
function chkreginfo(form,mark,edit){
if(mark==0 || mark=="all"){                      //验证收货人
if(form.taker.value==""){
chknew_taker.innerHTML="请输入收货人！";
form.taker.style.backgroundColor="#FF0000";
```



```
return false;
}else{
chknew_taker.innerHTML="";
form.taker.style.backgroundColor="#FFFFFF";
}
        }
        if(mark==1 || mark=="all"){                        //验证邮编
if(form.code.value==""){
chknew_code.innerHTML="请输入邮编！";
form.code.style.backgroundColor="#FF0000";
return false;
}else if(isNaN(form.code.value)){
chknew_code.innerHTML="邮编由数字组成！";
form.code.style.backgroundColor="#FF0000";
return false;
}else if(form.code.value.length!=6){
chknew_code.innerHTML="邮编由 6 位数字组成！";
form.code.style.backgroundColor="#FF0000";
return false;
}else{
chknew_code.innerHTML="";
form.code.style.backgroundColor="#FFFFFF";
}
}
if(mark==2 || mark=="all"){                                //验证电话号码
if(form.tel.value==""){
chknew_tel.innerHTML="请输入电话号码！";
form.tel.style.backgroundColor="#FF0000";
return false;
}else if(!checkregtel(form.tel.value)){
chknew_tel.innerHTML="电话号码的格式不正确！";
form.tel.style.backgroundColor="#FF0000";
return false;
}else if(isNaN(form.tel.value)){
chknew_tel.innerHTML="电话号由数字组成！";
form.tel.style.backgroundColor="#FF0000";
                      return false;
}else{
chknew_tel.innerHTML="";
form.tel.style.backgroundColor="#FFFFFF";
}
}
if(mark==3 || mark=="all"){                                //验证联系地址
if(form.address.value==""){
chknew_address.innerHTML="请输入联系地址！";
form.address.style.backgroundColor="#FF0000";
return false;
}else{
chknew_address.innerHTML="";
```

```
form.address.style.backgroundColor="#FFFFFF";
}
      }
}
```

8.6.2 动态生成订单号

动态生成订单号应用的是 time()函数，该函数获取当前的 UNIX 时间戳，返回值为从 UNIX 纪元（格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数。该函数的语法如下：

```
int time ( void )
```

time()函数没有参数，返回值为 UNIX 时间戳的整数值。

这里就是应用 time()函数返回的 UNIX 时间戳作为订单号的。在订单处理页 settle_chk.php 中，通过 time()函数获取 UNIX 时间戳的整数值，并且将其赋给$formid 变量作为订单号。关键代码如下：

```
$formid=time();                                    //生成订单号
```

8.6.3 WebBrowser 打印

在线支付模块中订单的打印和预览是非常重要的两个功能，实现这两个功能主要应用的是 WebBrowser 打印技术。打印预览的运行结果如图 8.16 所示。



图 8.16　打印预览的运行结果



WebBrowser 是 IE 内置的浏览器控件，无须用户下载。其优点是客户端独立完成打印目标文档的生成，减轻服务器负荷；缺点是源文档的分析操作复杂，并且对源文档中要打印的内容进行约束。

订单打印和预览主要应用 IE 内置的 WebBrowser 控件，该控件的具体参数如表 8.1 所示。

表 8.1　IE 内置的 WebBrowser 控件的参数说明

参　数	说　明
document.all.WebBrowser.Execwb(7,1)	表示打印预览
document.all.WebBrowser.Execwb(6,1)	表示打印
document.all.WebBrowser.Execwb(6,6)	表示直接打印
document.all.WebBrowser.Execwb(8,1)	表示页面设置
document.all.WebBrowser.Execwb(1,1)	打开页面
document.all.WebBrowser.Execwb(2,1)	关闭所有打开的 IE 窗口
document.all.WebBrowser.Execwb(4,1)	保存网页
document.all.WebBrowser.Execwb(10,1)	查看页面属性
document.all.WebBrowser.Execwb(17,1)	全选
document.all.WebBrowser.Execwb(22,1)	刷新
document.all.WebBrowser.Execwb(45,1)	关闭窗体无提示

8.6.4　支付宝支付

本模块中通过支付宝完成在线支付。要使用支付宝进行在线支付，必须先在支付宝中注册一个支付宝账户，然后下载支付宝提供的接口文件。通过支付宝提供的接口文件实现电子商务网站与支付宝的交互。

在支付宝提供的接口文件中，提供了完整的实例程序，只要依照实例程序中的操作方法，将电子商务网站中指定的数据提交到支付宝提供的数组参数中，然后支付宝会通过自定义的方法将数据提交到指定的网站中。

（1）在 forminfo.php 文件中，根据从数据库中读取的数据，定义支付宝的数组参数，其关键代码如下：

```
require_once("alipay_service.php");
require_once("alipay_config.php");
$parameter = array(
"service" => "trade_create_by_buyer",     //交易类型，必填实物交易等于 trade_create_by_buyer（需要填写物流）
"partner" =>$partner,                     //合作商户号
"return_url" =>$return_url,               //同步返回
"notify_url" =>$notify_url,               //异步返回
"_input_charset" => $_input_charset,      //字符集，默认为 GBK
"subject" =>$formarr[0]["commo_name"],    //商品名称，必填
"body" => $formarr[0]["commo_num"],       //商品描述，必填
"out_trade_no" => $ddnumber,              //商品外部交易号，订单号，必填，每次测试都需要修改
"logistics_fee"=>$yprice,                 //物流配送费用
"logistics_payment"=>'BUYER_PAY',     //物流配送费用付款方式：BUYER_PAY（买家支付）
```

```
"logistics_type"=>'EXPRESS',        //物流配送方式：POST（平邮）、EMS（EMS）、EXPRESS（其他快递）
"price" => $formarr[0]["total"],                          //商品单价，必填
"payment_type"=>"1",                                      // 默认为 1，不需要修改
"quantity" => "1",                                        //商品数量，必填
"show_url" => $show_url,                                  //商品相关网站
"seller_email" => $seller_email                           //卖家邮箱，必填
);
$alipay = new alipay_service($parameter,$security_code,$sign_type);
$link=$alipay->create_url();
$smarty->assign("link",$link);
```

（2）在 forminfo.tpl 模板页中创建超链接，完成支付操作，其关键代码如下：

```
<!--支付宝支付的接口操作，提交的数据-->
    <a href="{$link}">支付宝支付</a>
<!------------------------------------------------------->
```

说明 在本模块中，只是介绍 PHP 中电子商务网站与支付宝进行交互的方法，并且包括各种辅助功能的实现方法。至于涉及支付宝提供的参数和设置都是个人虚拟的，真实的参数将在实际的运作中由支付宝提供。在实际的运作中，需要向支付宝申请一个账户，然后使用由支付宝提供相应的参数就可以应用此功能了。

8.7 本章小结

本项目使用 Smarty、PDO 数据库抽象层和 Ajax 等目前的主流技术，操作一个在线支付模块。通过本章的学习，读者不仅可以了解在线支付模块的操作流程，而且可以熟悉购物车、订单及加密技术的开发思想。

第 9 章

留言本模块

（PHP+ MySQL 5.0 实现）

本章以构建一个适合于中小型企业应用的留言本为目的，全面介绍留言本的设计思路及流程。在介绍留言本开发过程的同时，对 PHP 中一些常用函数及编程技巧也进行了详细介绍，读者不仅可以对留言本的开发过程及思路有个全面掌握，而且还可以学习到许多 PHP 的编程技巧，从而全面提高个人的基础及编程能力。留言本的开发过程虽然较为简单，却涵盖了动态网站开发的大部分经典功能模块，所以只要熟练掌握本章的内容，并且能够灵活应用，就可以开发出大型的网上论坛、博客以及其他一些网络中较流行的网站。

通过阅读本章，可以学习到：

- **掌握留言本的设计思路及流程**
- **掌握 MySQL 数据库的设计方式和方法**
- **掌握 PHP 连接 MySQL 数据库的方法**
- **能够灵活运用常用的 SQL 查询语句**
- **掌握 PHP 实现数据分页显示的方法**

9.1 留言本模块概述

无论是应用程序还是 Web 项目，开发人员只有提前做好规划并能够清楚地掌握项目的设计思路及流程，才能开发出高效、完善的项目。本节将通过留言本的功能阐述、功能结构等方面介绍留言本的设计思路及实现过程。

9.1.1 模块概述

一个适合于个人或小规模企业使用的留言本模块，只需具有添加留言和删除留言功能即可。由于本章开发留言本是以适用于中小型企业为出发点，以使读者全面掌握留言本的设计流程及提高个人的编程能力为目的，所以本章所介绍的留言本除具有添加留言和删除留言的功能外，还具有用户注册、用户登录、编辑留言等功能。本章所介绍的留言本运行效果如图 9.1 所示。



图 9.1 留言本运行效果

9.1.2 功能结构

在开发程序前，先规划一下留言本模块的功能结构，这样在开发程序时可以保证思路清晰。留言本模块的功能结构如图 9.2 所示。



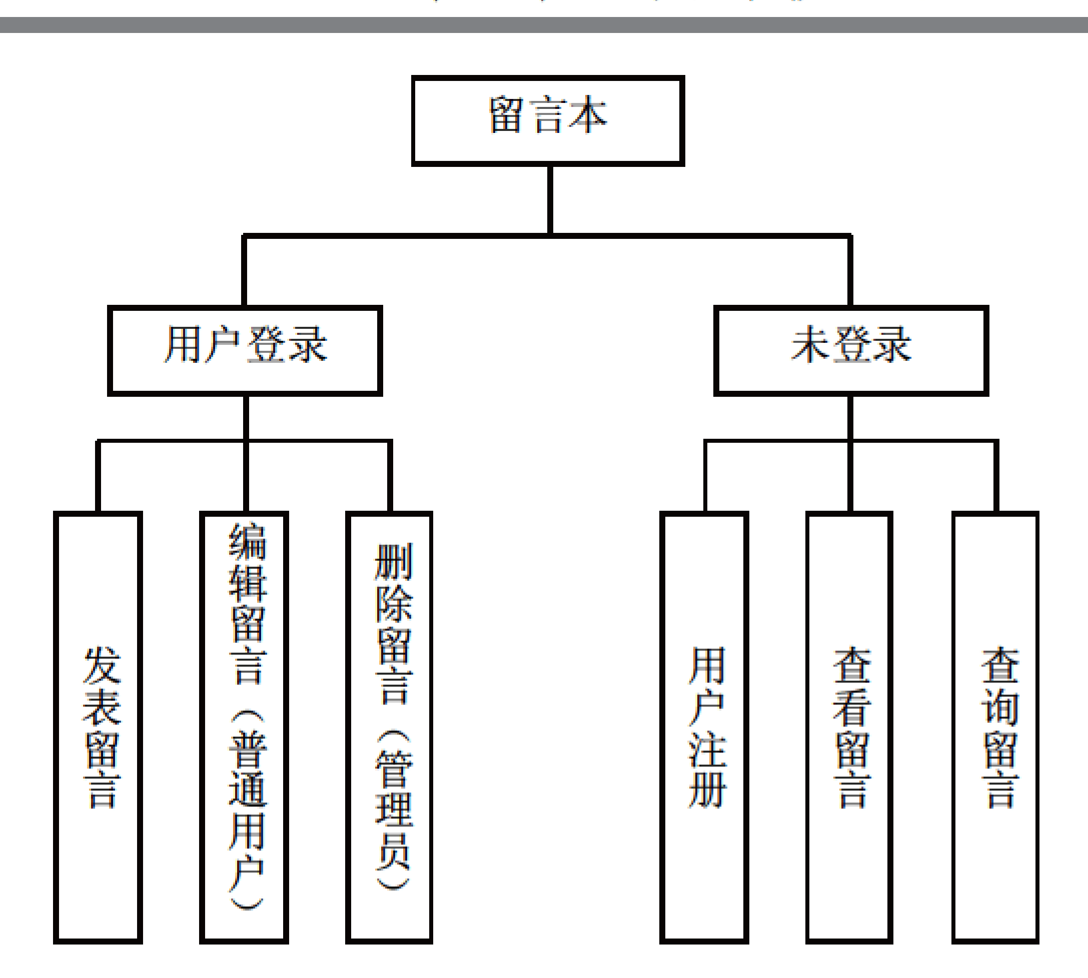


图 9.2　留言本模块的功能结构图

9.1.3　程序预览

留言本模块由多个功能模块组成，为了让读者对本模块有个初步的了解和认识，下面列出几个典型功能的页面，其他页面请参见光盘中的源程序。

留言本首页页面运行结果如图 9.3 所示。



图 9.3　留言本首页页面运行结果

用户注册页面运行结果如图 9.4 所示。用户只有注册成功后才可以获取登录用户名和密码，进行登



录后，才能执行发表留言等操作。

图 9.4　用户注册页面运行结果

发表留言页面运行结果如图 9.5 所示。查找留言页面运行结果如图 9.6 所示。

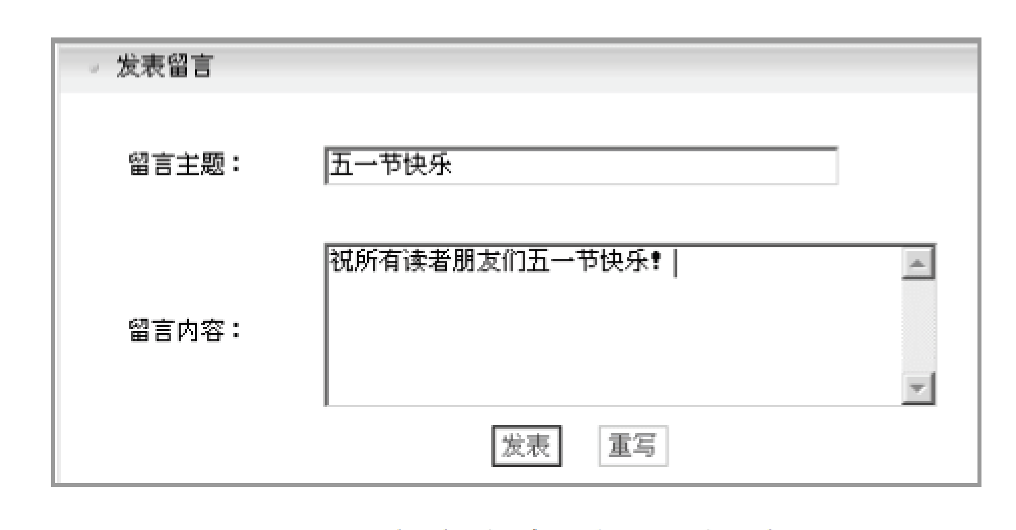


图 9.5　发表留言页面运行结果

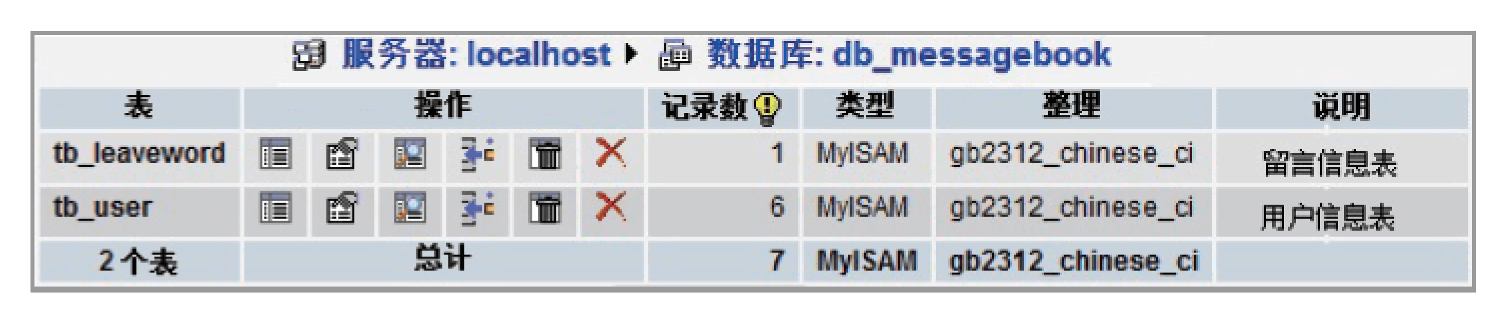


图 9.6　查找留言页面运行结果

9.2　数据库和数据表设计

9.2.1　数据库设计

留言本模块采用的是 MySQL 数据库，主要用于存储用户信息和留言信息。这里将数据库命名为 db_ messagebook，其中包含的数据表如图 9.7 所示。



服务器: localhost ▸ 数据库: db_messagebook

表	操作	记录数	类型	整理	说明
tb_leaveword		1	MyISAM	gb2312_chinese_ci	留言信息表
tb_user		6	MyISAM	gb2312_chinese_ci	用户信息表
2 个表	总计	7	MyISAM	gb2312_chinese_ci	

图 9.7　数据库结构



9.2.2　数据表设计

数据库设计完成后，下面来看各个数据表的结构和字段说明。

用户信息表主要用于存储用户信息，数据表字段设计如图 9.8 所示。

服务器: localhost ▸ 数据库: db_messagebook ▸ 表 : tb_user

字段	类型	整理	属性	Null	默认	额外	说明
id	int(8)			否		auto_increment	自动编号
userne	varchar(50)	gb2312_chinese_ci		否			用户名称
truename	varchar(50)	gb2312_chinese_ci		否			真实名称
email	varchar(50)	gb2312_chinese_ci		否			邮箱
qq	varchar(20)	gb2312_chinese_ci		否			QQ
tel	varchar(20)	gb2312_chinese_ci		否			电话
ip	varchar(10)	gb2312_chinese_ci		否			IP
address	varchar(250)	gb2312_chinese_ci		否			地址
face	varchar(50)	gb2312_chinese_ci		否			头像
regtime	datetime			否			发布时间
sex	varchar(2)	gb2312_chinese_ci		否			性别
usertype	int(2)			否			权限
userpwd	varchar(50)	gb2312_chinese_ci		否			登录密码

图 9.8　用户信息表

留言信息表主要用于存储用户留言信息，数据表字段设计如图 9.9 所示。



服务器: localhost ▸ 数据库: db_messagebook ▸ 表 : tb_leaveword

字段	类型	整理	属性	Null	默认	额外	说明
id	int(8)			否		auto_increment	留言id
userid	int(8)			否			留言者id
createtime	datetime			否			留言时间
title	varchar(250)	gb2312_chinese_ci		否			留言主题
content	text	gb2312_chinese_ci		否			留言内容

图 9.9　留言信息表

9.2.3　连接数据库

由于模块大部分页面都需要使用数据库，如果每页都编写相同的数据库连接代码，会显得十分繁琐。所以本模块将数据库连接代码单独存入一个文件 conn.php 中，在需要与数据库连接的页面中，使用包含语句包含 conn.php 文件即可。该模块实现与数据库连接的代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_ messagebook",$conn);              //选择数据库
mysql_query("set names gb2312");                       //设置页面编码格式
?>
```

9.3　发 表 留 言

9.3.1　发表留言概述

用户登录成功后，单击导航菜单中的“发表留言”超链接，进入发表留言页面，填写留言主题及

留言内容，单击“发表”按钮，即可发表留言。发表留言页面的运行结果如图 9.10 所示。



图 9.10 发表留言页面的运行结果

9.3.2 发表留言页面设计

一个优秀的 Web 程序，不仅应具有合理的代码编写规则和较高的代码执行效率，合理的页面设计方式和美观的页面也是不可缺少的。为了保证整个留言本页面的一致性，在设计页面时，将留言本的头部内容存储在 top.php 文件中，将用于显示版权信息的尾部内容存储在 bottom.php 文件中。这样，在新建留言本的功能页面时，只需在页面最上方加上 include_once("top.php")和在页面最下方加上 include_once("bottom.php")即可。应用这种页面设计方式，还可以提高程序的开发效率和易维护性。

技巧 为了提高网络的传输速度，尽量将页面图片存储为.gif 或.jpg 格式。另外，具有美观得体的页面也是开发人员所必须考虑的因素之一。

说明 之所以在设计页面时，将图片存储为.gif 或.jpg 格式，是因为这两种格式图片具有所占空间小、画面质量高等特点，从而可以有效提高网站的传输率。

发表留言页面的设计效果如图 9.11 所示。



图 9.11　发表留言页面的设计效果

发表留言页面的设计流程如下：

（1）应用 include 语句引用顶部 Banner 广告头文件 top.php 页，代码如下：

```
<?php include("top.php"); ?>
```

（2）应用 include 语句引用左侧功能导航文件 left.php 页，代码如下：

```
<?php include("left.php"); ?>
```

（3）为了使页面效果更丰富多彩、更人性化，分别定义文本输入框和按钮的 CSS 样式来制作个性化的表单。文本输入框和按钮的 CSS 样式代码如下：

```
<----------------------------定义按钮的 CSS 样式---------------------------->
buttoncss {
font-family: "Tahoma", "宋体";
font-size: 9pt; color: #FCC42C;
border: 1px #003399 solid;
color:006699;
BORDER-BOTTOM: #FCC42C 1px solid;
BORDER-LEFT: #FCC42C 1px solid;
BORDER-RIGHT: #FCC42C 1px solid;
BORDER-TOP: #FCC42C 1px solid;
```

```
background-color: #ffffff;
CURSOR: hand;
font-style: normal ;
}
<-----------------------------定义文本输入框的 CSS 样式---------------------------->
.inputcss {
font-size: 9pt;
font-family: "宋体";
font-style: normal;

}
```

发表留言页面涉及的 HTML 表单的重要元素如表 9.1 所示。

表 9.1　发表留言页面涉及的 HTML 表单的重要元素

名　称	类　型	含　义	重 要 属 性
form1	form	表单	name="form1" method="post" action="saveleaveword.php" onsubmit="return chkinput(this)"
title	text	留言主题	type="text" name="title" size="40" class="inputcss"
content	textarea	留言内容	name="content" rows="15" cols="55" class="inputcss"
submit	submit	“发表”按钮	type="submit" name="submit" value="发表" class="buttoncss"
reset	reset	“重写”按钮	type="reset" name="reset" value="重写" class="buttoncss"

9.3.3　将用户留言内容保存到数据库中

实现发表留言功能，首先应建立用于填写留言信息的表单，然后通过 PHP 脚本获取表单提交的数据，将留言信息添加到指定的数据表中。在 leaveword.php 中完成填写留言信息表单的创建，在 saveleaveword.php 文件中获取表单提交的留言信息，并且将其添加到指定的数据表中，其关键代码如下：

```
<?php
session_start();                         //初始化 Session 变量
include_once("conn.php");                //包含数据库连接文件
$sql=mysql_query("select id from tb_user where usernc='".$_SESSION["unc"]."'",$conn); //执行查询语句
$info=mysql_fetch_array($sql);           //获取查询结果
$userid=$info['id'];                     //获取用户 ID
$createtime=date("Y-m-d H:i:s");         //获取系统当前时间
//执行添加语句
if(mysql_query("insert into tb_leaveword(userid,createtime,title,content)
values('$userid','$createtime','".$_POST['title']."','".$_POST['content']."')",$conn)){
echo "<script>alert('留言发表成功！ ');history.back();</script>";
}else{
echo "<script>alert('留言发表失败！ ');history.back();</script>";
}
?>
```



9.4 查看留言

9.4.1 查看留言概述

单击导航菜单中的“查看留言”超链接，即可进入查看留言页面，其运行结果如图 9.12 所示。该页面展示出了留言主题、内容、发布者等信息，并且对当前登录用户的权限进行判断，如果当前用户是留言的发布者，那么还可以对留言进行编辑，否则只能查看留言内容。



图 9.12　查看留言页面运行结果

9.4.2 查看留言页面设计

为了使留言者和留言内容相互对应，在设计留言内容显示页面时，将留言者的常规信息动态显示在页面左侧，将留言内容显示在留言页面的右侧，并通过自定义函数对用户留言内容中的 HTML 标记进行过滤。查看留言页面的设计效果如图 9.13 所示。

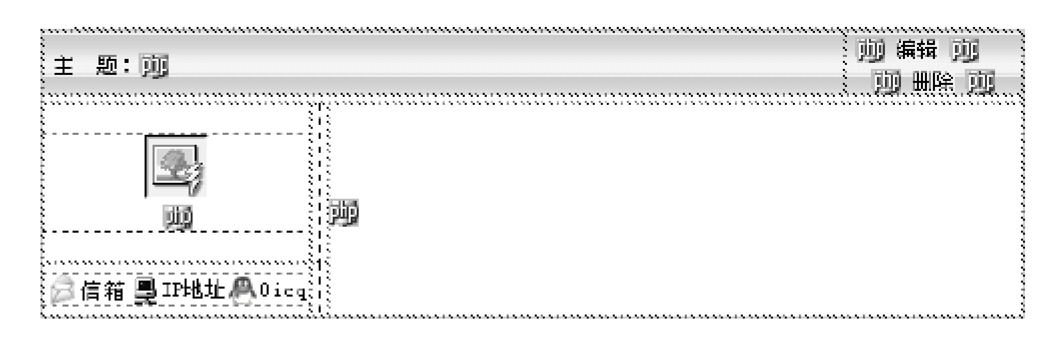


图 9.13　查看留言页面的设计效果

查看留言页面的设计流程如下：

（1）在 HTML 标记中嵌入 PHP 代码动态地显示用户常规信息，代码如下：

```
<img src="images/email.gif" width="45" height="16" alt="<?php echo $info1[email];?>"/>
<img src="images/ip.gif" width="55" height="16"   alt="<?php echo $info1[ip];?>"/>
<img src="images/qq.gif" width="45" height="16"   alt="<?php echo $info1[qq];?>"/>
```

（2）通过自定义函数 unhtml()实现用户留言中的 HTML 标记原样输出，代码如下：

```
function unhtml($content){
$content=htmlspecialchars($content);                    //将单引号、小于号、大于号等原样输出
$content=str_replace(chr(13),"<br>",$content);          //原样输出换行标记
$content=str_replace(chr(32)," ",$content);        //原样输出空格
return trim($content);
}
```

9.4.3 在页面中输出留言信息

在 index.php 文件中，通过 while 语句循环输出留言信息的内容，并且应用分页技术控制每页显示 3 条记录。关键代码如下：

```
<?php
$sql=mysql_query("select count(*) as total from tb_leaveword ",$conn);
$info=mysql_fetch_array($sql);
$total=$info[total];                                    //获取总留言条数
if($total==0){                                          //如果总留言条数为 0，则给出提示
echo "<div align=center>对不起，暂无留言！</div>";
}else{
//判断查询字符串 page 的值是否为空，如果为空，则默认显示第一页
if(!isset($_GET["page"]) || !is_numeric($_GET["page"])){
$page=1;
}else{
$page=intval($_GET["page"]);
}
$pagesize=3;                                            //规定每页显示 3 条留言
if($total%$pagesize==0){                                //获取总页数
$pagecount=intval($total/$pagesize);
}else{
   $pagecount=ceil($total/$pagesize);
}
$sql=mysql_query("select * from tb_leaveword order by createtime
desc limit ".($page-1)*$pagesize.",$pagesize   ",$conn);
while($info=mysql_fetch_array($sql)){                   //通过 while 循环显示所有留言

?>
…//显示用户信息和用户留言信息
<?php
$sqlu=mysql_query("select usernc from tb_user where id='".$info["userid"]."'",$conn);
$infou=mysql_fetch_array($sqlu);
```



```
if($infou["usernc"]==$_SESSION["unc"]){   //判断登录用户是否为留言的发表者，如果是则显示“编辑”按钮
?>
…//显示“编辑”按钮
<?php
}
?>
 
<?php
if($_SESSION["unc"]!=""){
$sqld=mysql_query("select usertype from tb_user where usernc='".$_SESSION["unc"]."'",$conn);
  $infod=mysql_fetch_array($sqld);
if($infod["usertype"]==1){       //判断当前登录用户是否为管理员，如果是则显示“删除”按钮
?>
<a href="deleteleaveword.php?id=<?php echo $info[id];?>" class="a1">删除</a>
<?php
}
}
?>
…//显示用户信息及用户留言信息
  <?php
}
}
?>
```

说明 上述代码中，实现留言信息的分页显示是通过 MySQL 数据库的扩展关键字 limit 来实现的，该关键字后跟两个参数。其中，第一个参数用于指定要显示记录的起始位置；而第二个参数用于指定所要显示的记录个数。

9.4.4　将留言信息进行分页显示

用户发表完留言后，可以通过查看留言页面查看所有留言内容。由于用户的留言数目较多，如果在同一页面中显示所有留言信息，则会给用户浏览带来很大的不便，所以通过分页的方式显示用户留言内容是不错的选择。在实现用户留言内容分页显示时，主要应用 is_numeric()函数判断用户通过 GET 方法提交的数据是否为数值型，并通过 ceil()函数对页码数据进行向上取整。下面将对这两个函数进行介绍。

1．is_numeric()函数

如果该函数的参数为数字或数字字符串，则返回 true，否则返回 false。其语法如下：

```
bool is_numeric ( mixed var )
```

参数 var 为要进行判断的数据。

2．ceil()函数

ceil()函数用于对浮点数进行向上取整。其语法如下：

```
float ceil ( float value )
```

参数 value 为要进行向上取整的数据。

在显示用户留言时，根据传入当前页面查询字符串 page 的值来决定所要显示的记录范围。创建分页超链接的关键代码如下：

```
<table width="550" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="351"><div align="left">共有留言 <?php echo $total;?> 条 
每页显示<?php echo $pagesize;?> 条 第 <?php echo $page;?> 
页/共 <?php echo $pagecount;?> 页</div>
</td>
<td width="199"><div align="right">
<a href="<?php echo $_SERVER['PHP_SELF']?>?page=1" class="a1">首页</a> 
<a href="<?php echo $_SERVER["PHP_SELF"]?>?page=
<?php
if($page>1)
echo $page-1;
else
echo 1;
?>
" class="a1">上一页</a> 
<a href="<?php echo $_SERVER["PHP_SELF"]?>?page=
<?php
if($page<$pagecount)
echo $page+1;
else
echo $pagecount;
?>
" class="a1">下一页</a> 
<a  href="<?php  echo  $_SERVER['PHP_SELF'];?>?page=<?php  echo  $pagecount;?>"  class="a1"> 尾 页
</a></div>
</td>
</tr>
</table>
```

9.5 编 辑 留 言

9.5.1 编辑留言概述

为了保证留言者的留言信息不被他人随意修改，留言用户只能对个人的留言内容进行修改，无权修改他人的留言内容。编辑留言页面的运行结果如图 9.14 所示。单击“编辑”超链接，打开编辑留言窗口，修改留言内容后单击“编辑”按钮，即可完成修改留言操作。





图 9.14　编辑留言页面的运行结果

9.5.2　编辑留言页面设计

为了使用户能够查看到原留言的内容，在设计编辑窗口时，采用一个弹出窗口作为编辑表单。编辑留言页面的设计效果如图 9.15 所示。

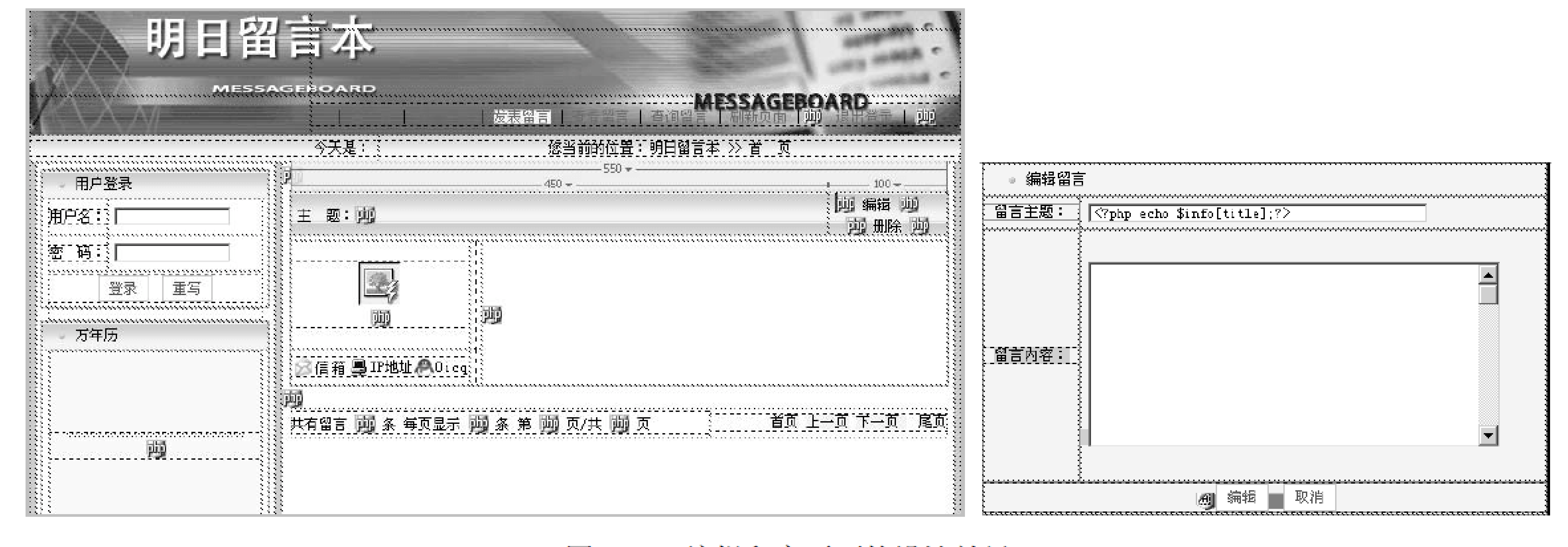


图 9.15　编辑留言页面的设计效果

编辑留言页面的设计流程如下：

（1）用户登录后，系统将数据库中所保存的用户信息与用户的登录信息进行比较，如果二者相同，则在该条留言后显示“编辑”按钮。代码如下：

```
<?php
$sqlu=mysql_query("select usernc from tb_user where id='".$info["userid"]."'",$conn);
$infou=mysql_fetch_array($sqlu);
if($infou["usernc"]==$_SESSION["unc"]){
?>
<a href="javascript:openeditwindow(<?php echo $info[id];?>)" class="a1">编辑</a>
<?php
}
?>
```

（2）当用户单击“编辑”按钮后，将通过自定义的 openeditwindow()方法打开一个新编辑窗口来编辑用户留言。代码如下：

```
<script language="javascript">
function openeditwindow(x){                    //定义 openeditwindow()方法打开弹出窗口
window.open("editleaveword.php?id="+x,"newframe","top=100,left=200,width=450,height=280,menubar=no,
location=no,scrollbars=no,status=no");
}
</script>
```

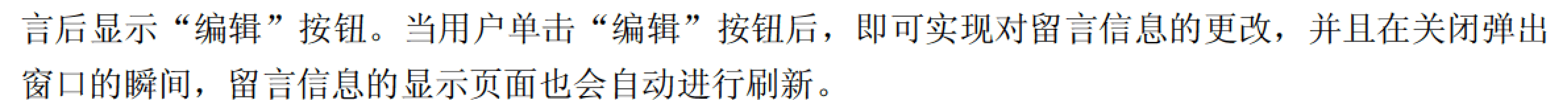
编辑留言页面涉及的 HTML 表单的重要元素如表 9.2 所示。

表 9.2　编辑留言页面涉及的 HTML 表单的重要元素

名　称	类　型	含　义	重 要 属 性
form1	form	表单	name="form1" method="post" action="<?php echo $_SERVER["PHP_SELF"];?>" onSubmit="return chkinput(this)"
title	text	留言主题	name="title" type="text" class="inputcss" size="45" value="<?php echo $info[title];?>"
content	textarea	留言内容	name="content" cols="52" rows="12" class="inputcss"
id	hidden	留言内容 id	type="hidden" name="id" value="<?php echo $_GET[id];?>"
submit	submit	“编辑”按钮	type="submit" value="编辑" class="buttoncss" name="submit"
reset	reset	“取消”按钮	type="reset" name="reset" value="取消" class="buttoncss"

9.5.3　编辑留言内容功能实现

为了保证用户的留言内容不被他人私自更改，在具体实现该功能时，采用每个用户只能对自己的留言内容进行更改的方式。其实现原理如下：首先，将标识用户身份的 Session 变量的值与数据库中该条留言对应的留言者进行比较，如果二者相同，则说明该条留言为当前登录用户所发表的，在该条留



言后显示“编辑”按钮。当用户单击“编辑”按钮后，即可实现对留言信息的更改，并且在关闭弹出窗口的瞬间，留言信息的显示页面也会自动进行刷新。

执行编辑留言操作的 editleaveword.php 文件的代码如下：

```
<?php
include_once("conn.php");
$id=$_GET["id"];
//判断是否单击“编辑”按钮，如果是执行如下用于实现用户留言信息更改的代码，并刷新父窗口和关闭弹出窗口
if($_POST["submit"]!=""){
if(mysql_query("update tb_leaveword set title='".$_POST["title"]."',content='".$_POST["content"]."'
where id='".$_POST["id"]."'",$conn)){
echo "<script>alert('留言更改成功！');window.opener.location.reload();window.close();</script>";
}else{
echo "<script>alert('留言更改失败！');history.back();</script>";
}
exit;
}
$sql=mysql_query("select * from tb_leaveword where id='".$id."'",$conn);
//如果用户未单击“编辑”按钮，则查询该用户的留言内容，并最终将未编辑的留言内容显示在表单中
$info=mysql_fetch_array($sql);
?>
```

说明 编辑用户留言，用户实现关闭弹出窗口，自动刷新父窗口，是通过如下语句实现的：

```
window.opener.location.reload();
```

上述代码的实现原理是：在用 window.close()语句关闭弹出窗口前，调用父窗口的 reload()方法实现父窗口的刷新。

9.6 删除留言

9.6.1 删除留言概述

如果用户发表非法留言，留言本管理员有权对其进行删除。管理员登录后将在每条留言后显示“删除”按钮，当管理员单击该按钮后，将弹出一个对话框提示管理员是否真正删除该条留言。如果单击对话框中的“确定”按钮，则该条留言将被删除，反之不做任何操作。

说明 本模块的管理员用户名为 mr，密码为 mrsoft。

删除留言页面的运行结果如图 9.16 所示。



图 9.16 删除留言页面的运行结果

9.6.2 删除留言页面设计

删除留言页面的设计效果如图 9.17 所示。

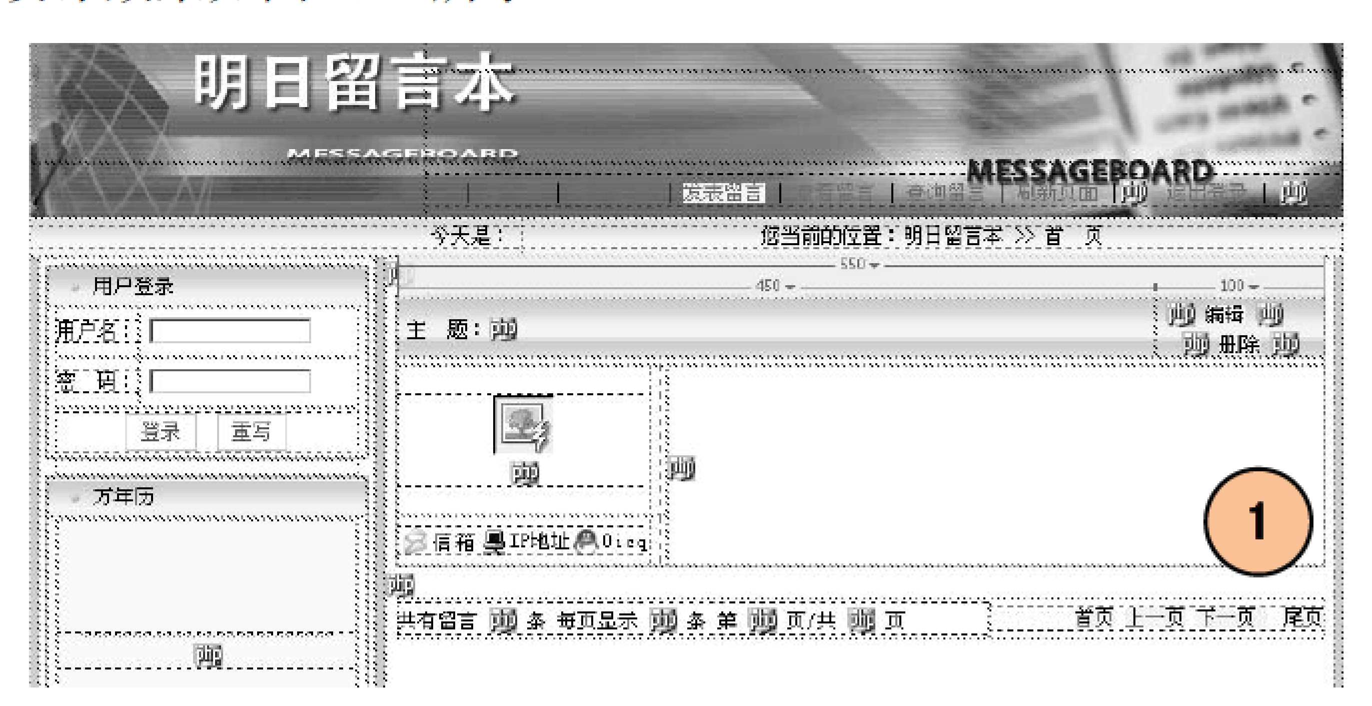


图 9.17 删除留言页面的设计效果

判断当前登录用户是否为管理员，如果是则显示“删除”按钮，代码如下：

```
<?php
if($_SESSION["unc"]!=""){                    //判断用户是否登录
$sqld=mysql_query("select usertype from tb_user where usernc='".$_SESSION["unc"]."'",$conn);
$infod=mysql_fetch_array($sqld);
if($infod["usertype"]==1){                   //判断登录用户是否为管理员
?>
```



```
<a href="javascript:if(window.confirm('确定删除该留言信息吗？')==true)
{window.location.href='deleteleaveword.php?id=<?php echo $info[id];?>';}" class="a1">删除</a>
//如果是则显示“删除”按钮
<?php
}
}
?>
```

9.6.3　删除留言内容功能实现

当用户确认删除留言后，将进入留言删除处理页 deleteleaveword.php 文件中，根据超链接传递的 id 值，执行删除操作。deleteleaveword.php 的代码如下：

```
<?php
include_once("conn.php");                    //包含数据库连接文件
if(mysql_query("delete from tb_leaveword where id='".$_GET["id"]."'",$conn)){
echo "<script>alert('留言删除成功！');history.back();</script>";
}else{                                        //删除指定的留言，并给出提示
echo "<script>alert('留言删除失败！');history.back();</script>";
}
?>
```

9.7　查 询 留 言

9.7.1　查询留言概述

随着留言本使用年限的增多，留言的数量会越来越多。为了方便浏览者的浏览和快速定位到指定的留言，在制作留言本时，设计了可以按留言主题和留言内容进行模糊查找、按留言者进行匹配查找的查询留言模块。

查询留言页面的运行结果如图 9.18 所示。



图 9.18　查询留言页面的运行结果

9.7.2　查询留言页面设计

查询留言页面的设计效果如图 9.19 所示。

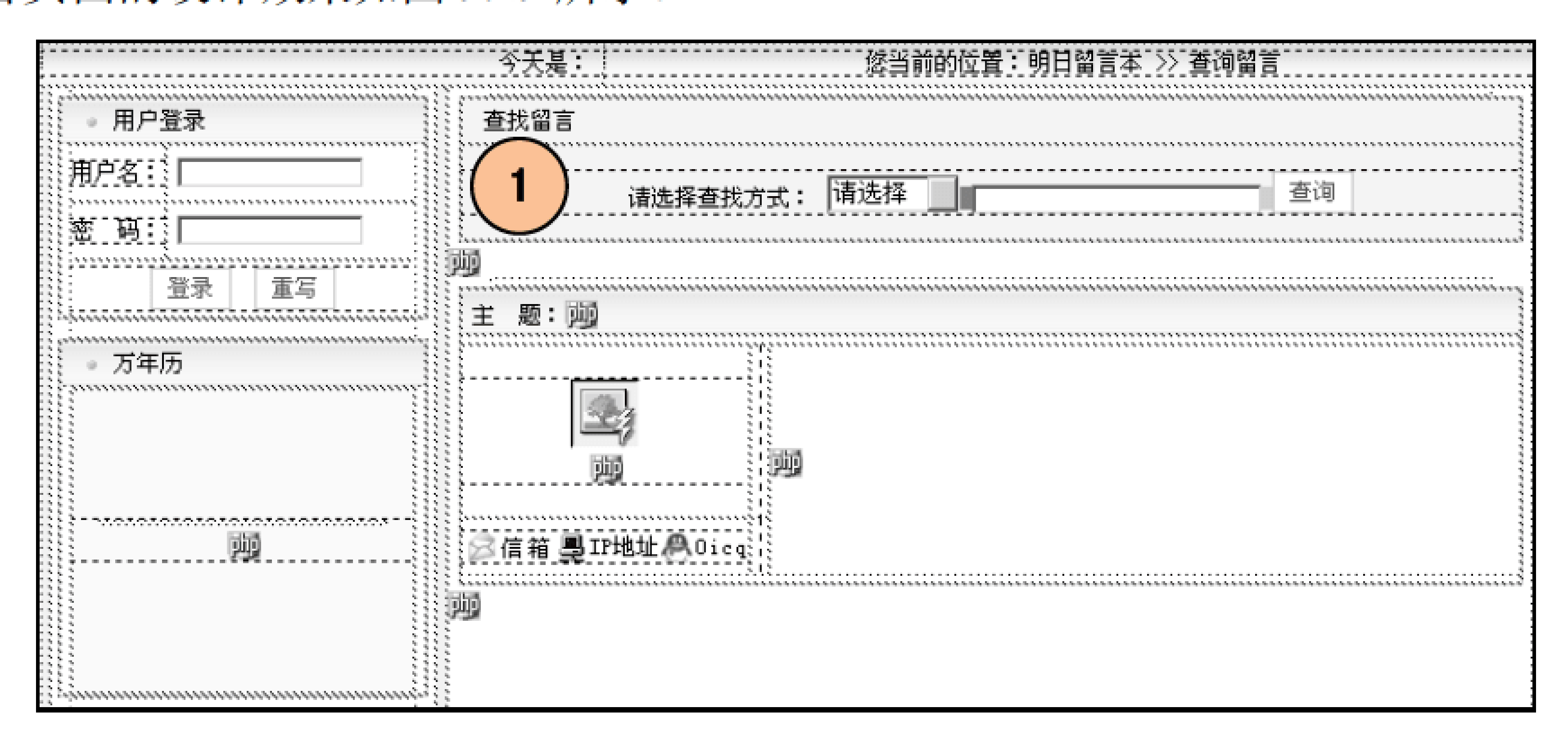


图 9.19　查询留言页面的设计效果

为了方便用户，所以设计了可以按多种方式进行查找的查询表单，代码如下：

```
<div align="center">请选择查找方式:
<select name="type">
<option value="">请选择</option>
<option value="1">主题</option>
<option value="2">内容</option>
<option value="3">留言者</option>
</select> <input type="text" name="keyword" size="25" class="inputcss"> 
<input type="submit" value="查询" class="buttoncss" name="submit">
</div>
```

查询留言页面涉及的 HTML 表单的重要元素如表 9.3 所示。

表 9.3　查询留言页面涉及的 HTML 表单的重要元素

名　称	类　型	含　义	重 要 属 性
form1	form	表单	name="form1" method="post" action="<?php echo $_SERVER["PHP_SELF"];?> " onsubmit="return chkinput_search(this)"
type	select	查询类型	name="type"
keyword	text	查询关键字	type="text" name="keyword" size="25" class="inputcss"
submit	submit	"查询"按钮	type="submit" value="查询" class="buttoncss" name="submit"

9.7.3　查询留言内容

当用户按要求添加完查找关键字，选择了查找方式后，单击"查询"按钮，数据将被提交到本页，在本页完成数据的查询操作，并且输出查询结果。searchword.php 的关键代码如下：



```
<?php
if($_POST["submit"]!=""){                //判断用户是否单击“查询”按钮，如果是则开始查询工作
$type=$_POST["type"];
$keyword=$_POST["keyword"];
if($type==1){                            //根据用户提交的查询类型定义查询语句
$sql=mysql_query("select * from tb_leaveword where title like '%".$keyword."%'",$conn);
}elseif($type==2){
$sql=mysql_query("select * from tb_leaveword where content like '%".$keyword."%'",$conn);
}elseif($type==3){
$sql0=mysql_query("select id from tb_user where usernc='".$keyword."'",$conn);
$info0=mysql_fetch_array($sql0);
$sql=mysql_query("select * from tb_leaveword where userid='".$info0["id"]."'",$conn);
}
$info=mysql_fetch_array($sql);
if($info==false){                        //判断是否查找到相关内容，如果没查找到，则显示提示信息
echo "<br><br><div align=center>对不起，没有查找到您要查找的内容！</div>";
}else{
do{                                      //通过循环显示查询结果
?>
…//显示查找内容
<?php
}while($info=mysql_fetch_array($sql));
}
}
?>
```

9.8　技 术 提 炼

9.8.1　将数据保存到数据库中

将用户发表的留言保存到 MySQL 数据库是通过 SQL 语言的 insert 语句实现的。PHP 通过 mysql_query()函数向 MySQL 数据库中发送 SQL 命令，所以 mysql_query()函数在数据库编程过程中具有极其重要的作用，下面将对该函数进行介绍。

mysql_query()函数用于向与指定的连接标识符所关联的数据库服务器中发送一条查询语句。其语法如下：

```
resource mysql_query (string query [, resource link_identifier])
```

参数 query 为必选参数，用于指定向数据库中发送的查询语句；link_identifier 为可选参数，是 mysql_connect()函数成功连接数据库后所返回的连接标识，如果省略了该参数，则使用最近一次与数据库建立连接后所返回的连接标识。

9.8.2　通过 JavaScript 实现弹出窗口

通过 JavaScript 实现弹出窗口，主要是通过调用 window 对象的 open()方法实现的，并将实现调用

的代码封装到一个单独的自定义方法中，最后通过 HTML 元素的 onclick 事件对该方法进行调用。

open()方法的语法如下：

```
open("url","name","features")
```

该方法的参数说明如表 9.4 所示。

表 9.4　open()方法的参数说明

参数取值	说　　明
url	可选参数。用于指定弹出新窗口的 URL 地址，如果省略该参数或其值是一个空字符串，那么新窗口就不显示任何文档
name	新窗口的名字，如果该名字是一个已经存在的窗口，那么 open()方法就不再创建一个新窗口，而只是返回这个窗口的引用，在这种情况下参数 features 将被忽略
features	主要是显示窗口的一些特征，该参数如果不设定，则显示这个窗口的所有特征。features 的常用特征请参见表 9.5

表 9.5　参数 features 的常用特征

名　　称	用　　法
channelmode	指定窗口是否应该以频道的方式显示，取值有 yes/no 或 1/0
fullscreen	指定窗口是否全屏显示，取值有 yes/no 或 1/0
menubar	指定窗口是否有菜单栏，取值有 yes/no 或 1/0
scrollbars	指定窗口是否有水平及竖直滚动条，取值有 yes/no 或 1/0
status	指定窗口是否有状态栏，取值有 yes/no 或 1/0
toolbar	指定窗口是否有工具栏，取值有 yes/no 或 1/0
location	指定窗口是否有地址栏，取值有 yes/no 或 1/0
directions	指定新建窗口是否有标准目录按扭，取值有 yes/no 或 1/0
resizable	指定新窗口大小是否可以调整，取值有 yes/no 或 1/0
top	以像素为单位指定新窗口上沿与屏幕上沿的距离
left	以像素为单位指定新窗口左沿与屏幕左沿的距离
width	以像素为单位指定新窗口的宽度
height	以像素为单位指定新窗口的高度

9.8.3　包含文件函数

在开发 Web 程序时，经常会在不同的页面中实现相同的功能或执行相同的代码。为了提高程序开发效率，可以将相同的代码存储到单独的文件中，然后用包含语句在页面中包含该文件即可。

include_once 语句用于在脚本执行期间包含并运行指定的文件。其语法如下：

```
include_once(filename)
```

参数 filename 用于指定要包含的文件。





9.8.4　MySQL 数据库的函数

PHP 中提供了一组操作 MySQL 数据库的函数，应用这些函数可以方便地对 MySQL 数据库进行管理。

mysql_fetch_array()函数获得数据库中满足函数 mysql_query()中的 SQL 语句的记录，其返回值是一个数组，该数组的下标可以是字段名，也可以是索引下标，数组元素的值是某个字段的内容。该函数会使记录指针自动向下移动，当移动到最后一行将返回一个 false 值。其语法如下：

```
array mysql_fetch_array(int result,int[result_type])
```

该函数的参数说明如表 9.6 所示。

表 9.6　mysql_fetch_array()函数的参数说明

参数取值	说　明
result	必选参数。mysql_query()函数向 MySQL 数据库发送查询命令的返回标识
result_type	可选参数。如果该参数取 MYSQL_BOTH，则该函数将返回一个同时包含关联索引和数字索引的数组；如果该参数取 MYSQL_ASSOC，则返回一个关联索引的数组；如果该参数取 MYSQL_NUM，则返回一个数字索引的数组

说明 与该函数功能类似的函数还有 mysql_fetch_rows()、mysql_result()、mysql_fetch_object()等。mysql_fetch_rows()函数返回数组的下标为数值索引下标。函数 mysql_result()有两个参数，其中第一个参数也是 mysql_query()的返回结果，而第二个参数可以是字段的偏移量或者字段名，一定注意它返回的结果不是数组，而是 MySQL 结果集中一个单元的内容。函数 mysql_fetch_object()的返回结果是一个对象，使用时只能通过字段名来返回结果。

9.9　本章小结

本章主要对留言本的设计思路和业务流程进行了介绍，并详细介绍了 PHP 连接和管理 MySQL 数据库的方法及留言本各模块的实现过程。通过本章的学习，不仅可以提高读者自身的基础，而且可以全面掌握适合于中小型企业应用的留言本的制作方法，最终可以在此基础上进行延伸，制作出更大规模的 Web 项目。

第10章

博客模块

（PHP+MySQL+mysqli 实现）

如果有人问"什么是Web 2.0？"，恐怕没几个人能说得清楚，但是要问什么是博客，哪怕是从不上网的人也是耳熟能详的了。博客正是Web 2.0概念中重要的组成部分之一（大家熟知的还包括IM即时通和RSS阅读器）。

Blog（博客），全名Weblog，后来缩写为Blog。Blogger就是写Blog的人，习惯于在网上写出日记、发布个人照片、展示个性自我的用户群体。对于Blog/Blogger的中文名称，有翻译成"博客"，也有翻译为"网志"，但大多数人都已经认可了"博客"。

通过阅读本章，可以学习到：

- 系统设计思路
- 数据库设计
- 博客空间个人首页设计
- 我的文章模块的设计
- 文章浏览模块设计
- 文章管理模块设计
- 好友管理模块设计
- 小纸条管理模块设计
- 用户管理模块设计



10.1　博客模块概述

10.1.1　模块概述

博客，即在网络上发表文章、展现个性的一类人，而博客系统，是为这类人群提供一个相互交流、学习的平台。博客的两大基本功能是共享与交流。共享，就是将文章、图片、心得等一些在传统概念中属于私人的东西，拿出来和多数人一起分享；交流，就是有着同样兴趣、爱好、语言的一类人之间的联系。

10.1.2　功能结构

本章的博客系统主要分为个人博客管理和博客后台管理两部分。

个人博客管理，主要功能包括个人管理、文章管理、相册管理、好友管理、留言管理和小纸条。个人博客管理功能结构如图 10.1 所示。

博客后台管理，主要是对博客用户和管理员的管理。博客后台管理包括用户管理、文章管理、相册管理、管理员管理、系统日志和数据备份。博客后台管理的功能结构如图 10.2 所示。

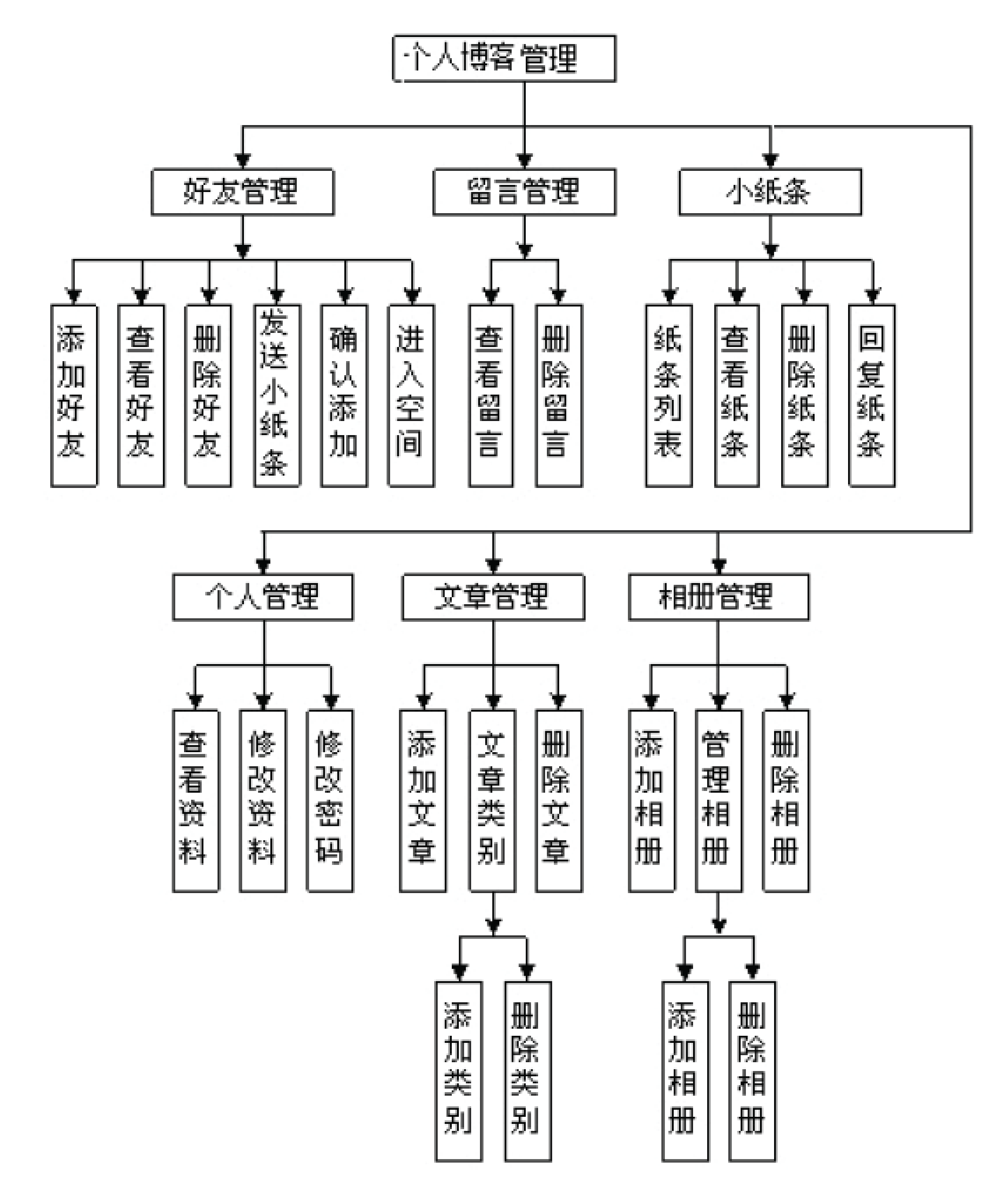


图 10.1　个人博客管理功能结构图

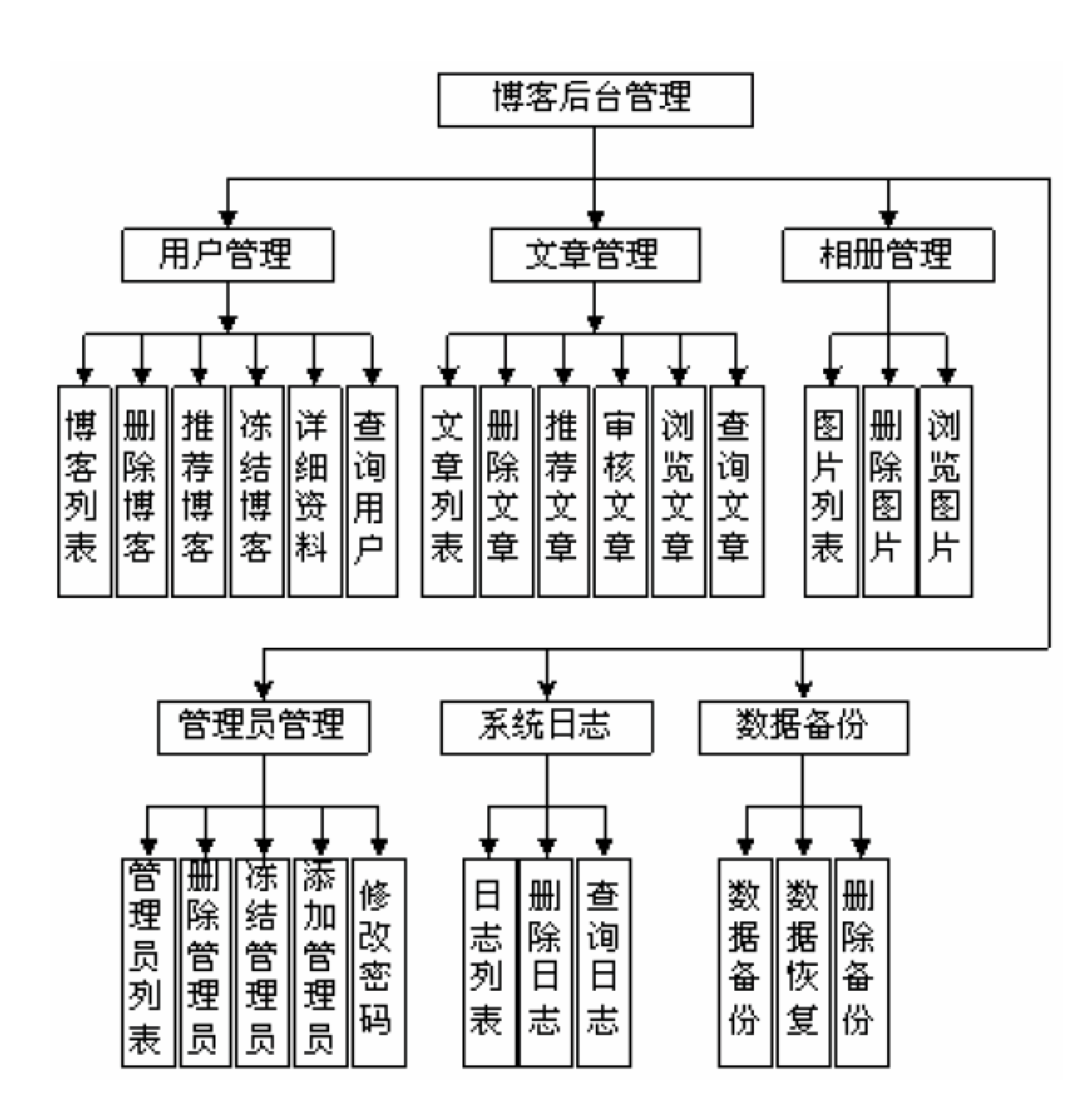


图 10.2　博客后台管理功能结构图

10.1.3　程序预览

为了让读者对本模块有个初步的了解和认识，下面给出本模块的几个页面运行效果图，如果想查

看完整的效果图，请参见光盘源程序。

博客首页页面运行效果如图 10.3 所示。



图 10.3　博客首页页面运行效果

个人博客管理页面运行效果如图 10.4 所示。



图 10.4　个人博客管理页面运行效果

博客用户管理页面运行效果如图 10.5 所示。

查询：编号 大于 查询

账号管理中心

	编号	用户账号	email	真实姓名	性别	生日	博客点击率	推荐	冻结/解冻	详细资料
□	25	leonsk			男	2011-01-01	0	否	冻结	详细资料
□	24	mrlx			男	2011-01-01	0	否	冻结	详细资料
□	23	lx111			男	2011-01-01	0	否	冻结	详细资料
□	22	a还珠格格	778452@qq.com	王佳丽	女	2011-01-01	3	否	冻结	详细资料
□	21	博客皇帝	1111111QQ@.com	张其卫	男	2011-01-01	6	推荐	冻结	详细资料
□	20	my BLOG	732565048QQ@.com	小丽	男	1925-10-30	5	否	冻结	详细资料
□	19	I LOVE YOU	845129549qq@.COM	123	女	2011-01-01	5	否	冻结	详细资料

全选 反选 删除选择　首页 上一页 下一页 尾页 当前是第1页/共1页7条记录 跳转到：1 页

技术服务热线：0431-84978981 传真：0431-84972266 企业邮箱：mingrisoft@mingrisoft.com
Copyright©www.mrbccd.com All Rights Reserved!

图 10.5　博客用户管理页面运行效果

图片管理页面运行效果如图 10.6 所示。



图 10.6　图片管理页面运行效果

10.2　数据库设计

本模块属于中小型个人网站，毫无争议，本模块采用的依然是 PHP+MySQL 这对黄金组合，无论

是从成本、性能、安全上考虑，还是从易操作性上考虑，MySQL 都是最佳选择。

10.2.1 创建数据库与数据表

博客系统使用的数据库是 db_blog，在数据库中一共使用了 9 个数据表。各个表的名称和作用如图 10.7 所示。

服务器: localhost ▸ 数据库: db_blog

表	操作	记录数	类型	整理	说明
tb_admin		1	MyISAM	gb2312_chinese_ci	管理员信息表
tb_article		10	MyISAM	gb2312_chinese_ci	博客文章列表
tb_frd		1	MyISAM	gb2312_chinese_ci	用户好友信息表
tb_log		127	MyISAM	gb2312_chinese_ci	文章日志信息表
tb_member		12	MyISAM	gb2312_chinese_ci	用户信息数据表
tb_mess		17	MyISAM	gb2312_chinese_ci	用户留言数据表
tb_review		31	MyISAM	gb2312_chinese_ci	文章评论表
tb_script		13	MyISAM	gb2312_chinese_ci	小纸条列表
tb_uppics		4	MyISAM	gb2312_chinese_ci	上传图片信息
9 个表	总计	216	MyISAM	gb2312_chinese_ci	

图 10.7 数据表列表

下面来具体看一下其中几张重要数据表的结构设计。

1．tb_article（博客文章列表）

tb_article 表包含类别名称、文章标题、文章内容等字段。tb_article 表结构及说明如图 10.8 所示。

服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_article

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
typename	varchar(20)	gb2312_chinese_ci		否			类别名称
title	varchar(50)	gb2312_chinese_ci		否			文章标题
content	mediumtext	gb2312_chinese_ci		否			文章内容
author	varchar(50)	gb2312_chinese_ci		否			文章作者
firsttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		发表时间
artquote	varchar(300)	gb2312_chinese_ci		否			是否引用
hitnum	int(4)			否	0		点击率
renum	int(4)			否	0		回复率
examine	int(1)			否	0		审核
isnominate	int(1)			否	0		是否推荐

图 10.8 tb_article 表结构及说明

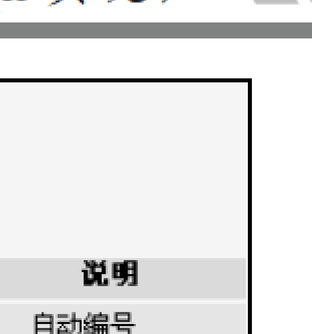
2．tb_frd（用户好友信息表）

tb_frd 表包含好友名称、添加人、好友等级和添加时间等字段。tb_frd 表结构及说明如图 10.9 所示。

3．tb_member（用户信息数据表）

tb_member 表包含用户的基本资料，如 E-mail、真实姓名和出生日期等，还有个人博客中的一些参数，如点击率、文章类别、图片类别、是否推荐等。tb_member 表结构及说明如图 10.10 所示。





服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_frd

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
frdname	varchar(100)	gb2312_chinese_ci		否			好友名称
frdmem	varchar(100)	gb2312_chinese_ci		否			添加人
frdlevel	int(1)			否			好友等级
addtime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		添加时间

图 10.9　tb_frd 表结构及说明

服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_member

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
name	varchar(50)	gb2312_chinese_ci		否			用户名称
pwd	varchar(50)	gb2312_chinese_ci		否			用户密码
question	varchar(50)	gb2312_chinese_ci		否			密保问题
answer	varchar(50)	gb2312_chinese_ci		否			密保答案
email	varchar(50)	gb2312_chinese_ci		否			Email
realname	varchar(50)	gb2312_chinese_ci		否			真实姓名
sex	char(1)	gb2312_chinese_ci		否			性别
birthday	date			否	2008-06-12		出生日期
tel	varchar(20)	gb2312_chinese_ci		否			电话
address	varchar(200)	gb2312_chinese_ci		否			地址
homepage	varchar(200)	gb2312_chinese_ci		否			个人主页
qq	varchar(10)	gb2312_chinese_ci		否			qq号码
unwrite	varchar(200)	gb2312_chinese_ci		否			个人签名
headgif	varchar(200)	gb2312_chinese_ci		否			头像
hitnum	int(4)			否	0		点击率
upfile	int(4)			否	0		上传文件
uppics	int(4)			否	0		上传图片
lasttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		最后登录
freeze	int(1)			否	0		是否冻结
blogname	varchar(100)	gb2312_chinese_ci		否			博客名称
blogurl	varchar(200)	gb2312_chinese_ci		否			博客链接
arttype	varchar(300)	gb2312_chinese_ci		否			文章类别
pictype	varchar(300)	gb2312_chinese_ci		否			图片类别
isnew	int(1)			否	0		是否新注册
isnominate	int(1)			否	0		是否推荐

图 10.10　tb_member 表结构及说明

4．tb_review 表（文章评论表）

tb_review 表包含被评论的文章 id、评论内容、评论人以及评论时间等字段。tb_review 表结构及说明如图 10.11 所示。

5．tb_uppics 表（上传图片信息）

tb_uppics 表包含上传图片的信息，如图片名称、上传用户、图片类别等字段。tb_uppics 表结构及说明如图 10.12 所示。

服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_review

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
artid	int(4)			否			文章id
content	varchar(300)	gb2312_chinese_ci		否			评论内容
man	varchar(100)	gb2312_chinese_ci		否			评论人
firsttime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		评论时间

图 10.11 tb_review 表结构及说明

服务器: localhost ▸ 数据库: db_blog ▸ 表 : tb_uppics

浏览 结构 SQL 搜索 插入 导出 Import 操作 清空 删除

字段	类型	整理	属性	Null	默认	额外	说明
id	int(4)			否		auto_increment	自动编号
picname	varchar(50)	gb2312_chinese_ci		否			图片名称
picpath	varchar(100)	gb2312_chinese_ci		否			图片路径
upauthor	varchar(100)	gb2312_chinese_ci		否			上传用户
pictype	varchar(50)	gb2312_chinese_ci		否			图片类别
hitnum	int(4)			否	0		点击率
examine	int(1)			否	0		是否审核
uptime	timestamp		ON UPDATE CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP		上传时间

图 10.12 tb_uppics 表结构及说明

此外，还有管理员信息表、文章日志信息表、用户留言数据表和小纸条列表。限于篇幅，这里不再介绍，读者可参见本书附赠光盘中的数据库文件。

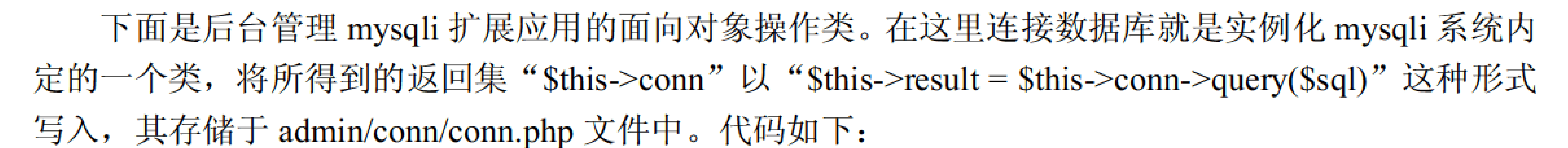
10.2.2 数据库操作类

为了方便使用，减少代码，本模块使用 mysqli 扩展存取数据，与 mysql 扩展相比，mysqli 扩展在以下方面有了明显的提高。

- ☑ 兼容性与维护性：mysqli 扩展可以很容易地使用 mysql 的新功能，所以 mysqli 拥有与 mysql 更高的兼容性。即使 mysql 的新版本又出现了更多功能，mysqli 扩展也可以很容易地支持。
- ☑ 面向对象：mysqli 扩展已封装到一个类中，从而可使用面向对象的方式编程。即使对面向对象不了解，mysqli 也可以应用面向过程的编程方式来实现。
- ☑ 速度和安全性：mysqli 扩展执行的速度比之前版本的 mysql 扩展快了很多。mysqli 扩展支持 mysql 新版本的密码杂凑（Passwoed Hashes）和验证程序，从而提高了应用程序的安全性。
- ☑ 预准备语句：可提高重复使用语句的性能，mysqli 扩展提供了对预准备语句的支持。
- ☑ 调试功能：mysqli 扩展进一步改进了调试功能，提高了开发效率。要在 PHP 中使用 mysqli 扩展，需要在配置文件 php.ini 中添加“extension=php_mysqli.dll”设置，如果配置文件中已有上述设置，确保 extension 前面没有“;”，否则将其去掉。

下面介绍如何使用 mysqli 扩展来存取数据库。mysqli 提供面向对象和面向过程两种方式来与数据库交互。在本模块中，后台管理应用面向对象方式与数据库交互，前台应用面向过程方式与数据库交互。





下面是后台管理 mysqli 扩展应用的面向对象操作类。在这里连接数据库就是实例化 mysqli 系统内定的一个类，将所得到的返回集“$this->conn”以“$this->result = $this->conn->query($sql)”这种形式写入，其存储于 admin/conn/conn.php 文件中。代码如下：

```
//定义私有属性，如主机名、用户名、用户密码、数据库名等
<?php
class opmysqli{
private $host = '127.0.0.1';                    //服务器地址
private $name = 'root';                         //登录账号
private $pwd = '111';                           //登录密码
private $dBase = 'db_blog';                     //数据库名称
private $conn = '';                             //数据库链接资源
private $result = '';                           //结果集
private $msg = '';                              //返回结果
private $fields;                                //返回字段
private $fieldsNum = 0;                         //返回字段数
private $rowsNum = 0;                           //返回结果数
private $filesArray = array();                  //返回字段数组
private $rowsArray = array();                   //返回结果数组
//使用__construct 关键字初始化类。这与 mysql 扩展写法一样
function __construct($host='',$name='',$pwd='',$dBase=''){
if($host != '')
$this->host = $host;
if($name != '')
$this->name = $name;
if($pwd != '')
$this->pwd = $pwd;
if($dBase != '')
$this->dBase = $dBase;
$this->init_conn();
}
```

连接数据库需要实例化 mysqli 类。该类是 PHP 环境自带的，所以不要管它从哪里来，只需实例化就可以。

```
function init_conn(){
$this->conn=new mysqli($this->host,$this->name,$this->pwd,$this->dBase);
$this->conn->query("set names gb2312");
}
```

下面是一些具体的操作方法，包括查询结果、取得字段数、更新、删除、添加记录数等。其部分代码如下：

```
//查询结果
function mysqli_query_rst($sql){
if($this->conn == ''){
$this->init_conn();
```

```
}
$this->result = @$this->conn->query($sql);
}
//取得字段数
function getFieldsNum($sql){
$this->mysqli_query_rst($sql);
$this->fieldsNum = @$this->result->num_fields($this->result);
}
//取得查询结果数
function getRowsNum($sql){
$this->mysqli_query_rst($sql);
$this->rowsNum = @$this->result->num_rows;
return $this->rowsNum;
}
//取得字段数
function getRowsArray($sql){
$this->mysqli_query_rst($sql);
while($row = @$this->result->fetch_array(MYSQLI_ASSOC)) {
$this->rowsArray[] = $row;
}
return $this->rowsArray;
}
//更新、删除、添加记录数
function uidRst($sql){
if($this->conn == ''){
$this->init_conn();
}
$this->conn->query($sql);
$this->rowsNum = @$this->conn->affected_rows;
return $this->rowsNum;
}
//最后实例化这个类即可
$conne = new opmysqli();
?>
```

前台使用 mysqli 面向过程与数据库进行交互。它的使用方法类似于 mysql 扩展。例如，mysql_query() 这个 mysql 扩展函数，在 mysqli 扩展中是写为 mysqli_query()。返回结果集在 mysql 中是 mysql_query ($sql，$conn)，而在 mysqli 扩展中则是 mysqli_query($conn，$sql)。下面给出 mysqli 面向过程操作数据库的部分代码，其存储于 center/conn/conn.php 文件中。

```
//连接数据库
function init_conn(){
$this->conn=mysqli_connect($this->host,$this->name,$this->pwd,$this->dBase);
mysqli_query($this->conn,"set names gb2312");
}
//查询结果
function mysqli_query_rst($sql){
```

```
if($this->conn == ''){
$this->init_conn();
}
$this->result = @mysqli_query($this->conn,$sql);
}
//取得字段数
function getFieldsNum($sql){
$this->mysqli_query_rst($sql);
$this->fieldsNum = @mysqli_num_fields($this->result);
}
```

有关数据库操作类的讲解完毕，其具体代码请参考本书光盘中的内容。

10.3　博客空间个人首页

10.3.1　博客空间个人首页概述

博客系统的模块较多，限于篇幅，这里只对重点模块进行讲解。首先来看个人首页。个人首页由多个页面组成，例如热门文章、文章分类、留言列表等。其页面的设计结构如图 10.13 所示。

<table>
<tr><td colspan="3">登录/注册（进入后台/退出登录）</td></tr>
<tr><td colspan="3">Banner</td></tr>
<tr><td>显示新消息</td><td colspan="2">导航栏（我的文章　我的相册　我的资料　我的好友）</td></tr>
<tr><td>博客信息
（personinfo.php）</td><td rowspan="4">内容显示区
（默认newart1.php）</td><td>热门文章列表
（hotart.php）</td></tr>
<tr><td rowspan="2">日历
（date.html）</td><td>文章类别列表
（arttype.php）</td></tr>
<tr><td>图片类别列表
（pictype.php）</td></tr>
<tr><td>留言列表
（mess1.php）</td><td>好友链接
（allfrd.php）</td></tr>
<tr><td colspan="3">留言板（lyb.php）</td></tr>
</table>

图 10.13　博客空间个人首页结构及布局

首页的运行效果如图 10.14 所示。





图 10.14　博客首页运行效果

10.3.2　首页模块实现过程

个人首页设计极其简单，总体是采用 div+css 样式设计的，例如导航条、个人首页的头部、外层等，都是采用 div+css 设计的。然后再使用包含文件结合技术，把所有需要显示在首页的文件和配置文件引入进来即可。具体步骤如下：

（1）在 center\center.php 文件中加载各种文件，例如配置文件、数据库类文件、计数统计文件、CSS 样式文件、js 脚本文件等。代码如下：

```
<?php
session_start();                                        //开启 Session
header('Content-Type:text/html;charset=gb2312');        //设置页面编码
include_once '../config.php';                           //加载配置文件
include_once 'conn/conn.php';                           //加载数据库类
include_once 'inc/count.php';                           //加载计数器
?>
<link href="../css/style.css" rel="stylesheet" type="text/css" />
<script language="javascript" src="../js/xmlhttprequest.js"></script>
<script language="javascript" src="../js/center.js"></script>
<script language="javascript" src="../js/date.js"></script>
<script language="javascript" src="../js/personinfo.js"></script>
…
```

（2）创建“注册登录”超链接。判断当前的 Session 值是否为空，如果为空，就显示“登录注册”

按钮；否则，显示光临信息和“进入管理页面”的超链接。代码如下：

```
<?php
if(!empty($_SESSION['name']) and !is_null($_SESSION['name'])){
?>
您好，欢迎光临<?php echo $_SESSION['name']; ?>的博客！
<a id='blogmain' onclick="javascript:open('manage.php','_parent','',false)">进入管理页面</a> 
<a onclick="javascript:open('logout.php','_parent','',false)">退出登录</a> 
<?php
}else{
?>
<a onclick="javascript:window.open('login.php','login','width=300,height=200',false)">登录</a>
<a onclick="javascript:window.open('register.php','register','',false)">注册</a> 
<?php
}
?>
```

（3）创建 Banner，并且输出博客空间的名称。Banner 下面的一行，由两部分构成。一部分是信息消息提醒；另一部分是导航栏。新消息提醒每隔一段时间，查看一下个人的纸条信息，如果发现新纸条，会马上显示出来。代码如下：

```
<div id="nav">
<div id="newscrip"> 
<script>setInterval("shownewmess()",10000);</script>
</div>
<div id="rightfloat">
<a onclick='javascript:open("center.php?uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我的文章</a>
| <a onclick='javascript:open("center.php?act=pics&uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我的相
册</a>
| <a onclick='javascript:open("center.php?act=person&uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我
的资料</a>
| <a onclick='javascript:open("center.php?act=frd&uid=<?php echo $_GET['uid'] ?>","_parent","",false)'>我的好
友</a></div>
</div>
```

（4）加载各个页面。其中，内容显示区是首页面的核心，几乎所有博客空间的功能都将在内容显示区显示。内容显示区的代码如下：

```
...
<?php
if(isset($_GET['act']) and $_GET['act'] == ''){
include_once 'newart1.php';
}else if(isset($_GET['act']) and $_GET['act'] == 'see'){
include_once 'showarticle1.php';
}else if(isset($_GET['act']) and $_GET['act'] == 'pics'){
include_once 'pics1.php';
```

```
}else if(isset($_GET['act']) and $_GET['act'] == 'person'){
include_once 'person.php';
}else if(isset($_GET['act']) and $_GET['act'] == 'frd'){
include_once 'frd1.php';
}else if(isset($_GET['act']) and $_GET['act'] == 'modart'){
include_once 'modart.php';
}else{
include_once 'newart1.php';
}
?>
</div>
…
```

10.4 我 的 文 章

10.4.1 我的文章概述

进入博客个人空间，或者单击博客个人空间的“我的文章”超链接，将在内容显示区显示博客中已存在的文章的简要信息列表，如图 10.15 所示。其中，包括文章标题、发表时间、部分文章内容和一个“阅读全文”超链接。



图 10.15 文章列表的运行结果

单击“阅读全文”超链接，打开 showarticle1.php 页面，将显示要查看的文章的全文和评论信息。同时，还显示相关的一些操作，例如引用文章和发表评论。如果是本人的博客，登录后还可以修改文章、删除文章和删除评论，如图 10.16 所示。





图 10.16　浏览文章页面的运行结果

10.4.2　我的文章功能实现过程

创建 newart1.php 文件，完成我的文章模块设计。首先设置分页信息，然后生成 SQL 查询语句，用来查找当前用户所发表过的文章，并返回查询结果。实现代码如下：

```
<?php
include_once 'conn/conn.php';
if(isset($_GET['uid'])){
$num = 7;                                                          //每页显示记录数
if(!isset($_GET['curpage']) ){
$curpage = 1;
}else{
$curpage = $_GET['curpage'];                                       //当前页
}
$sql = "select * from tb_article where author = '".$_GET['uid']."' order by id desc ";
$totnum = $conne->getRowsNum($sql);                                //记录数
$totpage = ceil($totnum / $num);                                   //页数
$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;            //SQL 查询语句
$arr = $conne->getRowsArray($tmpsql);                              //获取数组
$conne->close_rst();
?>
```

如果获取到的数组不为空，那么循环判断是否为引用，如果不是，则直接输出文章信息；如果是，那么根据引用的文章 id 号来查找原文，并输出文章信息。最后，添加一个“阅读全文”超链接。代码如下：

```
if($arr != ''){
foreach($arr as $value){
if(!empty($value['artquote']) and !is_null($value['artquote'])){
$tmparr = explode(',',$value['artquote']);
$sql1 = "select * from tb_article where id = ".$tmparr[0];
$artarr1 = $conne->getRowsArray($sql1);
$conne->close_rst();
…
}else{
…
}
}
```

10.4.3 文章浏览功能实现过程

文章浏览页（showarticle1.php）最核心的功能当然是浏览文章。首先，根据超链接传递的文章 id，将文章信息全部显示出来。代码如下：

```
<?php
include_once '../config.php';                                //载入配置文件
include_once 'conn/conn.php';                                //载入数据库类
$showartsql = "select * from tb_article where id = ".$_GET['artid'];   //生成查询语句
$artarr = $conne->getRowsArray($showartsql);                 //返回数组形式的记录
$conne->close_rst();                                         //清空结果集
if(!empty($artarr[0]['artquote']) and !is_null($artarr[0]['artquote'])){   //判断文章是否为引用
$tmparr = explode(',',$artarr[0]['artquote']);               //得到引用的来源地址
$sql1 = "select * from tb_article where id = ".$tmparr[0];  //根据地址查询数据
$artarr1 = $conne->getRowsArray($sql1);                      //返回结果集
$conne->close_rst();
…
}else{
…
}
?>
```

然后，实现文章引用（其实质是引用文章地址）。当其他的博客用户看到好文章时，可以通过“引用”超链接，将文章 id 保存到自己的文章列表中，通过对文章 id 的访问，可以省去手工输入的麻烦，也在一定程度上节省了表空间。此外，如果博主想对自己的这篇文章进行宣传，也可以将引用地址复制下来，粘贴到其他地方来提高博客空间的人气。实现引用的代码如下：

```
[<a onclick="return artquote('<?php echo $_GET['uid'] ?>','<?php echo $_GET['artid']; ?>')" style="color:
#FF6600;">引用</a>] 来自动添加到您的博客空间<br />
<input id="copyadd" onclick="copyaddress()" type="text" size="50" value="<?php echo 'http://'.$_SERVER['HTTP_
HOST'].$_SERVER['REQUEST_URI']; ?>" readonly="readonly" style="border: 1px #FFFFFF solid;"   />
```

该段代码主要通过两个 JavaScript 自定义函数来控制。第 1 个是 artquote()函数，将博客用户、文章 id 传给 showarticle_chk.php 页，根据返回结果进行处理；第 2 个是 copyaddress()函数，主要实现复制当前页面 URL 地址的功能。脚本文件 js\center.js 的代码如下：



```
//引用
function artquote(uid,quoteid){
xmlhttp.open('get','showarticle_chk.php?act=quote&uid='+uid+'&quoteid='+quoteid,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
msg = xmlhttp.responseText;
if(msg == '1'){
alert('引用成功');
}else if(msg == '2'){
alert('您还没有登录');
}else if(msg == '3'){
alert('引用失败');
}else if(msg == '4'){
alert('不允许引用自己的文章');
}else{
alert(msg);
}
}
}
xmlhttp.send(null);
}
//自动复制网址
function copyaddress(){
	$('copyadd').select();
	document.execCommand('COPY');
	alert('成功复制本地址');
}
```

showarticle_chk.php 页处理和文章相关的操作，包括添加评论、删除评论和引用。主要代码如下：

```
<?php
session_start();                                          //开启 Session
header('Content-Type:text/html;charset=gb2312');          //设置编码
include_once 'conn/conn.php';                             //载入数据库类
$artid = $_GET['artid'];                                  //文章 id
$quoteid = $_GET['quoteid'];                              //引用 id
$name = $_SESSION['name'];                                //当前用户
$uid = $_GET['uid'];                                      //原文章用户
$act = $_GET['act'];                                      //何种操作
$reback = '0';
if(empty($act)){
if($name != $uid){
$selectsql = "select renum from tb_article where id = ".$artid;
$renum = $conne->getFields($selectsql,0);
$upsql = "update tb_article set renum = ".(++$renum)." where id = ".$artid;
$num = $conne->uidRst($upsql);
}
$content = $_GET['cont'];
$sql = "insert into tb_review (artid,content,man) values('".$artid."','".$content."','".($name == "?'匿名':$name)."')";
$num = $conne->uidRst($sql);
$reback = '1';
}else if($act == 'quote'){                                //如果是引用操作
```

```
if(!empty($name) and !is_null($name)){
if($_GET['uid'] == $name){                              //如果是引用了自己的文章
$reback = '4';
}else{
$chkquote = $conne->getRowsNum("select artquote from tb_article where author = '".$name."'");
if($chkquote >= 1){
$reback = '5';
}else{
$selectsql = "select renum from tb_article where id = ".$quoteid;
$renum = $conne->getFields($selectsql,0);
$upsql = "update tb_article set renum = ".++$renum." where id = ".$quoteid;
$num = $conne->uidRst($upsql);
$oldid = $conne->getFields("select artquote from tb_article where id = ".$quoteid,0);
if(!empty($oldid) and !is_null($oldid)){
$saveid = $oldid.','.$quoteid;
}else{
$saveid = $quoteid;
}
$sql = "insert into tb_article (typename,author,artquote) values('日记','".$name."','".$saveid."')";
$num = $conne->uidRst($sql);
if($num == 1){
$reback = '1';
}else{
$reback = '3';
}
}
}
}else{
$reback = '2';
}
}else if($act == 'delone'){
$delonesql = "delete from tb_review where id= ".$_GET['rid'];
$num = $conne->uidRst($delonesql);
$reback = '1';
}else if($act == 'delall'){
$delallsql = "delete from tb_review where artid = ".$_GET['aid'];
$num = $conne->uidRst($delallsql);
$reback = '1';
}
echo $reback;
?>
```

10.5 文 章 管 理

10.5.1 文章管理概述

对一个博客系统来说，文章管理是最基本的功能，但也是最复杂的一个功能。文章管理模块包括发表文章、查看文章、删除文章、文章类别等功能，文章管理中发表文章的运行结果如图 10.17 所示。





图 10.17　发表文章页面的运行结果

10.5.2　发表文章功能实现过程

在发表文章页 addart.php 中，创建一个发表文章的表单，包括文章标题、文章类别、文字编辑、文章内容等表单元素。该页面的运行结果如图 10.17 所示。页面中的部分表单元素如表 10.1 所示。

表 10.1　发表文章页面的主要元素属性及说明

元 素 名 称	元 素 属 性	说　明
text	type="text" id="txt_title" size="68"	文章标题

select	<?php
	　　foreach($typearr as $value){
	?>
	　　<option value="<?php echo $value; ?>"><?php echo $value; ?></option>
	<?php
	}
	?>
		文章类别

select	name="font" class="wenbenkuang" id="font" onChange="showfont(this.options[this.selectedIndex].value)"	文章字体
select	class="wenbenkuang" onChange="showsize(this.options[this.selectedIndex].value)"	字体大小
select	onChange="showcolor(this.options[this.selectedIndex].value)" name="color" size="1" class="wenbenkuang" id="select"	字体颜色
textarea	cols="75" rows="20" id="file" style="border:0px;width:520px;"	文章内容
button	id="enterbtn" onClick="return chkart()"	“提交”按钮

当用户填写完博客主题和文章内容后，单击“提交”按钮，将触发 onclick 事件调用 chkart()函数。该函数将文章标题、文章类别等相关消息进行验证后，传递给 addart_chk.php 页，根据该页的返回值进

行不同的处理。chkart()函数存储于 js\article.js 脚本文件中，其关键代码如下：

```
function chkart(){
if($('txt_title').value==""){                                           //检测博客标题
alert("博客标题名称不允许为空！");
$('txt_title').focus();
return false;
}
if($('file').value==""){                                                //检测文章内容
alert("文章内容不允许为空！");
$('file').focus();
return false;
}
url = 'article/addart_chk.php?act=add&title='+$('txt_title').
value+"&arttype="+$('articletype'). value+"&cont=" +$('file').value;     //生成 URL
xmlhttp.open('get',url,true);
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4){
if(xmlhttp.status == 200){
msg = xmlhttp.responseText;                                             //获取返回值
if(msg == '1'){                                                         //如果等于 1，发表成功
alert('文章发表成功');
xmlhttp.open('get','article/article.php',true);                         //刷新文章列表
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4 && xmlhttp.status == 200){
$('showmenu').innerHTML = xmlhttp.responseText;
}}
xmlhttp.send(null);
}else if(msg == '2'){
alert('添加失败');
}else{
alert(msg);
}}}}
xmlhttp.send(null);
}
```

addart_chk.php 处理页接到文章信息后，将新信息添加到数据库中，如果成功，则返回 1。该页的代码如下：

```
<?php
session_start();                                                        //开启 Session
header('content-type:text/html;charset=gb2312');                        //设置编码
include "../Conn/conn.php";                                             //载入数据库类
$title=$_GET['title'];                                                  //文章标题
$typename = $_GET['arttype'];                                           //文章类别
$author=$_SESSION['name'];                                              //当前用户
$content=$_GET['cont'];                                                 //文章内容
$reback = '0';
$sql="Insert Into tb_article (typename,title,content,author)
```



```
Values ('".$typename."','".$title."','".$content."','".  $author."')";          //insert 语句
$num = $conne->uidRst($sql);                                                    //添加文章
if($num == 1){                                                                  //如果添加成功
$selectsql = "select upfile from tb_member where name = '".$author."'"
$artnum = $conne->getFields($selectsql,0);                                      //得到上传文章数量
$upsql = "update tb_member set upfile = ".(++$artnum)." where name = '".$author."'";
$num = $conne->uidRst($upsql);                                                  //更新
$reback = '1';
}else{
$reback = '2';
}
echo $reback;
?>
```

10.5.3　文章列表功能实现过程

文章列表页列出了文章的类别、标题和点击率。该页实现的功能包括全选、反选、删除和无刷新分页。文章列表页面的运行结果如图 10.18 所示。



图 10.18　文章列表页面的运行结果

文章列表页 article.php 的代码如下：

```
<?php
session_start();                                                  //开启 Session
header('content-type:text/html;charset=gb2312');                  //设置编码
include_once '../conn/conn.php';
if(isset($_GET['act'])){                                          //判断 act 参数
$act = $_GET['act'];
}else{
$act="";
}
$name = $_SESSION['name'];                                        //获取当前用户
$num = 10;                                                        //每页显示记录数
if(isset($_GET['curpage'])){                                      //判断是否为当前页
$curpage = $_GET['curpage'];
}else{
```

```
$curpage ="";
}
if($act == 'del'){                                                      //如果是删除操作
$rd = $_GET['rd'];                                                      //得到要删除的 id 序列
$delsql = "delete from tb_article where id = -1 ";                      //初始化删除语句
if(!empty($rd)){                                                        //如果删除序列不为空
$delarr = explode(',',$rd);                                             //拆分序列转为数组
foreach($delarr as $value){                                             //循环数组
$delsql .= "or id = ".$value." ";                                       //更新删除语句
}
$delnum = $conne->uidRst($delsql);                                      //执行删除语句
}
}
if(empty($curpage) or is_null($curpage)){                               //如果是第一次打开该页
$curpage = 1;                                                           //将当前页变量设为 1
}
if(empty($sql) or is_null($sql)){                                       //如果 SQL 查询语句为空
//创建新的查询语句
$sql = "select id,typename,title,hitnum,renum from tb_article where author = '".$name."' order by id desc";
}
$totnum = $conne->getRowsNum($sql);                                     //总记录数
$totpage = ceil($totnum / $num);                                        //总页数
$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;                 //分页
$arr = $conne->getRowsArray($tmpsql);                                   //得到记录数组
?>
```

上述代码实现了删除数据及分页使用的功能，接下来就是文章列表的显示。代码如下：

```
<div id = 'showarticle'>
<table width="600" border="1" cellpadding="0" cellspacing="0" align="center">
<tr>
<td height="25" width="30" align="center" valign="middle"> </td>
<td height="25" width="100" align="center" valign="middle">类别</td>
<td height="25" width="370" align="center" valign="middle">标题</td>
<td height="25" width="100" align="center" valign="middle">留言/点击率</td>
</tr>
<?php
foreach($arr as $key => $value){
?>
<tr>
<td align="center" valign="middle" height="25">
<input id="chk[<?php echo $key; ?>]" name="chk[]" type="checkbox" value="<?php echo $value['id']; ?>" />
</td>
<td align="center" valign="middle" height=25> <?php echo $value['typename']; ?></td>
<td align="left" valign="middle" height=25> <?php echo $value['title']; ?></td>
<td align="center" valign="middle" height=25> <?php echo $value['renum'].'/'.$value['hitnum']; ?> </td></tr>
<?php
}
```





以下是全选、反选、删除及分页的实现。删除则是全选、反选操作的延续。实现代码如下：

```
if($totnum > 0){                              //如果有记录
?>
<tr>
<td colspan="4" height="30" align="right" valign="middle">
<a href="#" onclick="return alldel('<?php echo count($arr); ?>')">全选</a>
<a href="#" onclick="return overdel('<?php echo count($arr) ?>')">反选</a>
<!--删除操作，调用 del()函数，并传递了两个参数-->
<button id="delbtn" name="delbtn" class="btn" onclick="return del('<?php echo $_SERVER['SCRIPT_ NAME']; ?>','
<?php echo count($arr); ?>',<?php echo $curpage; ?>)">删除选择</button>
<!--翻页超链接，将调用 pagination()函数-->
<?php echo ($curpage == 1)?"首页":"
<a   onclick='return pagination(\"".$_SERVER['SCRIPT_ NAME']." \",1)'>首页</a>"; ?>
<?php echo ($curpage==1)?"上一页":"
<a onclick='return pagination(\"".$_SERVER ['SCRIPT_NAME']. "\",".($curpage-1).")'>上一页</a>"; ?>
<?php echo ($curpage == $totpage)?'下一页':"
<a onclick='return pagination(\"".$_SERVER ['SCRIPT_ NAME']."\",".($curpage+1).")'>下一页</a>"; ?>
<?php echo ($curpage==$totpage)?"尾页":"
<a onclick='return pagination(\"".$_ SERVER['SCRIPT_ NAME']."\",".($totpage).")'>尾页</a>"; ?>
当前是第<?php echo $curpage; ?>页/共<?php echo $totpage; ?>页<?php echo $totnum; ?>条记录
<!--跳转操作，调用 jumpmem()函数-->
跳转到：<select id="jump" name="jump" onchange="return jumpmem('<?php echo $_SERVER['SCRIPT_
NAME']; ?>')">
<?php
for($i=1;$i<=$totpage;$i++){
        if($i == $curpage)
echo "<option value=".$i." selected>".$i."</option>";
else
echo "<option value=".$i.">".$i."</option>";
}
?>
</select> 页</td>
</tr>
<?php
}
?>
</table>
</div>
```

删除、翻页和跳转操作分别调用了脚本文件 choose.js 中的 3 个函数。删除函数 del()有 3 个参数，这 3 个参数分别代表删除处理页的 URL、当前页的记录总数和当前页的页码。

函数中首先声明了一个数组 rd，然后循环判断当前页中的复选框状态，将要删除的复选框的 vlaue 值保存到 rd 中；接着将 rd 和当前页的页码传给处理页；最后调用 rerst()函数将处理页的输出结果显示到指定区域。choose.js 文件的删除代码如下：

```
//删除所选
function del(chkurl,tot,curpage){
```

```
var rd=new Array();                                              //声明数组
for(i=0,j=0;i<tot;i++){                                          //循环判断复选框
if(!$('chk['+i+']').checked){
}else{                                                           //如果复选框被选中
rd[j] = $('chk['+i+']').value;                                   //将 value 值保存到 rd 数组中
j++;
}
}
if(rd == ''){                                                    //如果没有删除，提示错误
alert('请选取要删除的数据!');
return false;
}
url = chkurl+"?act=del&rd="+rd+'&curpage='+curpage;              //生成 URL
xmlhttp.open('get',url,true);                                    //创建新请求
xmlhttp.onreadystatechange = rerst;                              //调用 rerst()函数
xmlhttp.send(null);
return false;
}
function rerst(){
if(xmlhttp.readystate == 4){
if(xmlhttp.status == 200){                                       //判断请求是否完成
$('showarticle').innerHTML = xmlhttp.responseText;               //将返回值显示到指定区域
}
}
}
```

对翻页的处理统一使用 pagination()函数，该函数有两个参数：一个是处理页的 URL；另一个是当前页的页码。代码如下：

```
//分页
function pagination(chkurl,curpage){
url = chkurl+'?act=next&curpage='+curpage;                       //生成 URL
xmlhttp.open('get',url,true);                                    //创建新请求
xmlhttp.onreadystatechange = rerst;                              //调用 rerst()函数
xmlhttp.send(null);
return false;
}
```

跳转是翻页的另一种形式。翻页只能一页一页地翻，而跳转可以直接跳到指定的页面。代码如下：

```
//跳转
function jumpmem(chkurl){
jumppage = $('jump').value;                                      //获取跳转页数
url = chkurl+'?curpage='+jumppage;                               //生成 URL
xmlhttp.open('get',url,true);                                    //创建请求
xmlhttp.onreadystatechange = rerst;                              //调用 rerst()函数
xmlhttp.send(null);
return false;
}
```



10.5.4　文章类别功能实现过程

文章类别模块实现文章类别的添加、删除操作。文章类别页面的运行结果如图 10.19 所示。



图 10.19　文章类别页面的运行结果

文章的类别被保存到用户信息中，默认有一个“日记”类别，当有多个类别时，将使用逗号分割。显示文章类别时，首先获取该用户的类别字段的值，将字符串使用逗号分割成数组，最后由 foreach 循环显示所有类别。文章类别处理文件 arttype.php 的代码如下：

```
<?php
session_start();                                                //开启 Session
header('content-type:text/html;charset=gb2312');                //设置编码
include_once '../conn/conn.php';                                //载入数据库类文件
$act = $_GET['act'];                                            //获取 act 值
$name = $_SESSION['name'];                                      //获取用户名称
$sql = "select arttype from tb_member where name = '".$name."'";   //生成查询语句
$typefields = $conne->getFields($sql,0);                        //获取字段值
?>
<table id="addtypefm" width="300" border="1" cellspacing="0" cellpadding="0" align="center">
<tr>
<td colspan="2" height="25" align="center" valign="middle">文章类别管理</td>
</tr>
<?php
$typearr = explode(',',$typefields);                            //使用逗号分割字符串存为数组
foreach($typearr as $value){                                    //循环输出数组
?>
<tr>
<td width="200" height="25" align="center" valign="middle"> <?php echo $value; ?></td>
<td height="25" align="center" valign="middle"> 
<button id="deltype" value="<?php echo $value; ?>" class="btn"
onclick="return delarttype('<?php echo $typefields; ?>','<?php echo $value;?>')">删除</button></td>
</tr>
<?php
}
?>
```

```
<tr>
<td width="200" height="25" align="center" valign="middle">
<input id="addarttype" name="addarttype" type="text" class="txt" /></td>
<td height="25" align="center" valign="middle"><button id="addtype" value="<?php echo $value; ?>"
class="btn"
onclick="return addarttype('<?php echo $typefields; ?>')">添加新类</button></td>
</tr>
</table>
```

添加文章类别的操作是用户单击“添加新类”按钮后，将调用 addarttype()函数，并将原类别字段传给该函数。addarttype()函数将创建一个新请求，并根据请求的输出值做不同的处理。addarttype()函数存储于 js\article.js 脚本文件中，其代码如下：

```
//添加文章类别
function addarttype(typestr){
if($('addarttype').value == ''){                          //判断类别名称是否为空
alert('请输入添加类别名称');
$('addarttype').focus();
return false;
}
url = 'article/arttype_chk.php?act=add&typestr='+typestr+'&typename='+$('addarttype').value ;
xmlhttp.open('get',url,true);                             //创建新请求
xmlhttp.onreadystatechange = function(){
if(xmlhttp.readystate == 4){
if(xmlhttp.status == 200){
msg = xmlhttp.responseText;                               //获取处理页输出结果
//根据返回的不同结果，做不同的处理
if(msg == '2'){
alert('添加失败');
}else if(msg == '3'){
alert('添加名称重复');
}else if(msg == '1'){
showarttype();
}else{
alert(msg);
}
}
}
}
xmlhttp.send(null);
}
```

arttype_chk.php 是 addarttype()函数调用的处理页，将获取到的原类别字段后面加上逗号和新类别名称，再将该字段保存。代码如下：

```
<?php
session_start();                                          //开启 Session 支持
header('Content-Type:text/html;charset=gb2312');          //设置编码
include_once '../../conn/conn.php';                       //载入数据库类
```



```
$act = $_GET['act'];                                         //获取 act 值
$typestr = $_GET['typestr'];                                 //原类别字符串
$typename = $_GET['typename'];                               //要获取的类别字符串
$arr = explode(',',$typestr);                                //将原字符串转为数组
$reback = '0';
$name = $_SESSION['name'];                                   //获得当前用户
if($act == 'add'){                                           //如果是添加操作
if(in_array($typename,$arr)){                                //判断名称是否重复
$reback = '3';
}else{
$typestr .= ','.$typename;                                   //将新类别名称添加到原类别后面
$sql = "update tb_member set arttype = '".$typestr."' where name = '".$name."'";
$num = $conne->uidRst($sql);                                 //执行更新语句
if($num == 1){
$reback = '1';
}else{
$reback = '2';
}
}
}
…
```

删除文章类别和添加类别大同小异，但要注意两点：第一不允许删除默认的“日记”类别，如果删除的是“日记”类别，则弹出警告；第二将删除的类别下的所有文章，全部放到默认的“日记”目录下。删除文章类别操作由 arttype_chk.php 文件完成，其代码如下：

```
//添加类别代码
else if($act == 'del'){
if($typename == '日记'){
$reback = '3';
}else{
$tmpstr = '日记';
for($i=1;$i<count($arr);$i++){
if($arr[$i]  != $typename){
$tmpstr .= ','.$arr[$i];
}
}
$changsql = "update tb_article set typename = '日记'where author = '".$name."' and typename='". $typename."'";
$num = $conne->uidRst($changsql);
$sql = "update tb_member set arttype = '".$tmpstr."' where name = '".$name."'";
$num = $conne->uidRst($sql);
if($num == 1){
$reback = '1';
}else{
$reback = '2';
}
}
}
echo $reback;
```

10.6 用户管理

10.6.1 用户管理概述

用户管理模块是后台的一个主要模块，它是用来管理注册用户的，有删除用户、冻结用户、查看博客点击率、查看所有用户信息等多项功能。用户管理模块的运行结果如图 10.20 所示。



图 10.20　用户管理模块的运行结果

用户管理模块是博客后台系统所包含的 6 个模块之一。其他的模块依次为文章管理模块、图片管理模块、管理员设置模块、系统日志模块以及数据备份模块。

用户管理模块中包含的功能有查询博客、推荐博客、冻结博客、查看详细资料和删除博客等功能。其中推荐博客和冻结博客的实现方法基本相同，而查看详细资料和删除博客在前面也接触过，这里只讲解查询博客和推荐博客的实现方法。

10.6.2 用户管理功能实现过程

首先来看一下用户管理模块的首页 member.php。在 member.php 文件中，将数据库中存储的用户信息进行分页输出，并且完成各种附加功能的设置。其运行结果如图 10.21 所示。

	编号	用户账号	email	真实姓名	性别	博客点击率	推荐	冻结/解冻	详细资料
☐	25	leonsk			男	0	否	冻结	详细资料
☐	24	mrlx			男	0	否	冻结	详细资料
☐	23	lx111			男	0	否	冻结	详细资料
☐	22	a还珠格格	778452@qq.com	王佳丽	女	3	否	冻结	详细资料
☐	21	博客皇帝	1111111QQ@。com	张其卫	男	6	推荐	冻结	详细资料
☐	20	my BLOG	732565048QQ@.com	小丽	男	5	否	冻结	详细资料
☐	19	I LOVE YOU	845129549qq@.COM	123	女	5	否	冻结	详细资料

全选 反选 删除选择　　首页 上一页 下一页 尾页 当前是第1页/共1页7条记录 跳转到: 1 页

图 10.21　用户管理页面的运行结果



member.php 页的代码如下：

```
<?php
session_start();                                          //开启 Session
include_once '../conn/conn.php';                          //载入数据库类
?>
<script language="javascript" src="../js/choose.js"></script>  //载入 js 文件
<?php
$num = 10;                                                //每页显示记录数
if(empty($_GET['act'])){                                  //获取当前是第几页
$act = "";
}else{
$act = $_GET['act'];                                      //当前页
}
$sql = "";                                                //SQL 语句
if(empty($curpage) or is_null($curpage)){                 //获取当前是第几页
$curpage = 1;
}else{
$curpage = $_GET['curpage'];                              //当前页
}
if(empty($fields) or is_null($fields)){                   //排列顺序，默认为 id
$fields = 'id';
}else{
$fields = $_GET['fields'];                                //依照该字段排列记录
}
//是否有查询内容，如果没有
if(!isset($_REQUEST['cont'])){
//直接生成 SQL 查询语句，使用$fields 字段作为排序依据
$sql = "select id,name,email,realname,sex,birthday,hitnum,freeze,isnominate from tb_member order by
".$fields." desc";
//如果有查询内容，就获取要查询的字段、查询方式和查询内容
}else{
$querymem = $_REQUEST['querymem'];
$signslt = $_REQUEST['signslt'];
$cont = $_REQUEST['cont'];
//初始化变量$sql 为一个无条件的查询语句
$sql = 'select id,name,email,realname,sex,birthday,hitnum,freeze,isnominate from tb_member ';
//根据查询字段所在的组，生成不同的 where 子句
if(in_array($querymem,array('id','upfile','uppics','hitnum','birthday'))){
$sql .= " where ".$querymem." ".$signslt." ".$cont;
}else if(in_array($querymem,array('name','realname','qq','email'))){
if($signslt == 'exac'){
$sql .= "where ".$querymem." = '".$cont."'";
}else if($signslt == 'mist'){
$sql .= "where ".$querymem." like '%".$cont."%'";
}
}
//添加 order by 子句
$sql .= " order by ".$fields." desc";
```

```
}
$totnum = $conne->getRowsNum($sql);                    //记录数
$totpage = ceil($totnum / $num);                       //页数
$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;
$arr = $conne->getRowsArray($tmpsql);
?>
```

10.6.3 查询博客功能实现过程

查询博客是根据博客不同的信息来查询特定的博客。例如 id 号查询、点击率查询等。查询页是单独的一页，被包含在 member.php 页中。当用户查询时，该页将查询字段、查询方式和查询内容传给 member.php 页，再由 member.php 页进行处理，该页的处理过程在上面已经介绍过。查询页 query.php 的代码如下：

```
<?php
include_once '../conn/conn.php';                       //载入数据库类
?>
<script language="javascript" src="../js/query.js"></script>
<table border="0" cellpadding="0" cellspacing="0" align="center">
<form id="queryfm" name="queryfm" method="post" action="member.php?act=query" onSubmit="return chkquery()">
<tr>
<td width="50" height="25">查询：</td>
<td width="100">
<select id="querymem" name="querymem" onChange="return changequery()">
<option value="id">编号</option>
<option value="name">账号</option>
<option value="realname">真实姓名</option>
<option value="email">email</option>
<option value="qq">qq</option>
<option value="birthday">出生日期</option>
<option value="upfile">文章数</option>
<option value="uppics">上传图片</option>
<option value="hitnum">点击率</option>
</select>
</td>
<td width="50" align="center" valign="middle"><div id="sign">
<select id="signslt" name="signslt">
<option value="&gt;">大于</option>
<option value="=">等于</option>
<option value="&lt;">小于</option>
</select>
</div></td>
<td width="100"><div id="input"><input id="cont" name="cont" type="text" size="30" /></div></td>
<td><input id="querybt" name="querybt" type="submit" value="查询"></td>
</tr>
</form>
</table>
```



在查询页中，当用户选择不同的查询条件时，查询方式也会跟着发生变化。例如查询 id、点击率时，查询条件有大于、等于和小于；当查询账号、真实姓名、E-mail 时，查询条件变为精确和模糊。这是通过 query.js 脚本来控制的，实现代码如下：

```
function changequery(){
key = document.getElementById('querymem').value;        //获取查询条件
switch(key){
/*当查询条件和数字有关系时*/
case 'id':
case 'upfile':
case 'uppics':
case 'hitnum':
document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "&gt;">大于</Option><option value="=">等于</option><option value="&lt;">小于</option></select>';
document.getElementById('cont').value='';
break;
/*当查询条件和字符串有关系时*/
case 'name':
case 'realname':
case 'email':
case 'qq':
case 'title':
case 'author':
case 'typename':
case 'upname':
case 'upauthor':
case 'operator':
document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "exac">精确</Option><option value="mist">模糊</option></select>';
document.getElementById('cont').value='';
break;
/*当查询条件和时间有关系时*/
case 'birthday':
case 'firsttime':
case 'uptime':
document.getElementById('sign').innerHTML = '<select id="signslt" name="signslt">
<Option value= "&gt;">大于</Option><option value="=">等于</option><option value="&lt;">小于</option></select>';
document.getElementById('cont').value='查询格式：YYYY-MM-DD';
break;
/*省略部分为其他模块的查询方式设置*/
…
}
document.getElementById('cont').focus();
document.getElementById('cont').select();
}
```

10.6.4 推荐博客功能实现过程

当用户为推荐博客时，该用户的博客空间将在首页的显著位置显示，这样可以提高网站和个人空

间的人气和访问量。推荐博客的实现方法非常简单，在用户表中设置一个字段，名称是 isnominate，类型为 int。当用户注册时，字段的默认值为 0，当设为推荐博客时，将该字段设为 1。该操作从 member.php 文件中设置的超链接开始，其代码如下：

```
…
<a href="member_chk.php?act=nominate&id=<?php echo $value['id']; ?>&isnominate=<?php echo $value
['isnominate']; ?>"><?php echo ($value['isnominate']==0?"否":"推荐"); ?></a>
…
```

链接到 member_chk.php 文件，根据博客用户的 id 号和 isnominate 字段的值来更新用户表，代码如下：

```
<?php
session_start();
header('Content-Type:text/html;charset=gb2312');
include_once '../conn/conn.php';
$act = $_GET['act'];
if($act == 'del'){
…                                                     //删除操作
}else if($act == 'nominate'){                         //推荐博客
$id = $_GET['id'];                                    //获取用户 id 号
$nominate = ($_GET['isnominate']+1) % 2;              //将获取的字段值加 1 再取模
$sql = "update tb_member set isnominate = ".$nominate." where id = ".$id;
$cont = '推荐用户操作，账号：'.$id;                    //系统日志内容
$num = $conne->uidRst($sql);                          //更新字段值
if(!empty($num)){
$inssql = "insert into tb_log values('','".$cont."','".$_SESSION['manager']."',now())";
$num = $conne->uidRst($inssql);                       //添加日志
}
echo "<script>location='member.php';</script>";
}
```

10.7 技术提炼

10.7.1 文件包含技术

通过文件包含技术实现文件的包含，有助于网站页面的规划、管理和更新。例如，网站的头文件、尾文件和数据库的连接文件，都可以通过包含的方式进行载入，这样就可以避免在每个页面中编写相同的代码，即减少代码的冗余，同时也便于对这些内容的修改和更新。

在博客模块的个人首页中通过 include_once()函数包含文件。使用 include_once()函数在包含文件时，如果该文件中的代码已经被包含，则不会被再次包含。include_once()函数应用于脚本执行期间同一个文件有可能被包含超过一次的情况下，以确保它只被包含一次，从而避免出现函数重定义、变量重新赋值等问题。





博客个人首页的左侧包含文件代码如下：

```
<!--左侧-->
<div id="leftfloatcont">
<div id="personinfo">
<?php include_once 'personinfo.php'; ?>
</div>
<div id="date">
<?php include_once 'date.html'; ?>
</div>
<div id="showmessdiv">
<?php include_once 'mess1.php'; ?>
</div>
</div>
```

10.7.2　SQL 查询技术

在我的文章模块中，总体是采用 SQL 语句查询技术，从数据库中根据查询条件完成数据的循环输出。其应用到的 SQL 技术如下：

1．排序功能

所谓的排序功能就是先来后到之说，排序分为升序和降序，升序是从小到大，而降序则相反，是从大到小。排序主要有两个关键字：升序 asc 和降序 desc。其语法格式如下：

```
select * from  表名  order by 字段名  asc/desc
```

2．查记录总数

查记录总数就是将数据库中的记录总数查询出来。查询记录使用的关键字是 count 和 as。其语法如下：

```
select count (*) as (自定义字段) from  表名
```

3．条件查询

所谓的条件查询就是指定其中的一条或多条查询，它是带条件的，这里所用的关键字是 where。其语法格式如下：

```
select * from  表名  where  字段=条件
```

下面讲解一下这些 SQL 查询技术在本模块中的具体应用。

（1）我的文章中应用的查询语句的代码如下：

```
$sql = "select * from tb_article where author = '".$_GET['uid']."' order by id desc ";
$totnum = $conne->getRowsNum($sql);
$totpage = ceil($totnum / $num);
$tmpsql = $sql." limit ".($num *($curpage-1)).",".$num;
```

```
$arr = $conne->getRowsArray($tmpsql);
$conne->close_rst();
```

查询文章表（tb_article）中的所有记录，查询条件是 author 的值与超链接传递的值相同，并以 id 为条件进行降序排列。最后输出查询结果，完成我的文章标题、发表时间、部分文章内容的循环输出。

（2）文章浏览中应用的查询语句的代码如下：

```
$showartsql = "select * from tb_article where id = ".$_GET['artid'];          //生成查询语句
$artarr = $conne->getRowsArray($showartsql);                                    //返回数组形式的记录
$conne->close_rst();                                                            //清空结果集
```

以超链接中传递的文章的 id 为条件，执行查询操作，查询的结果就是有关这条记录的详细信息，即文章的详细内容。

10.7.3 会话处理技术

想要使用文章管理模块，前提是用户必须登录，匿名用户是无法访问这些功能的；想要删除文章和评论，前提是当前用户要么是管理员权限，要么是文章拥有者，否则不会显示删除功能。这两方面的控制都需要 Session 的配合，本节就来讲一下 Session 的应用及常见的问题处理。

Session 的中文译名为“会话”，是指用户从进入网站开始，直到关闭网站这段时间内，所有网页共同使用的公共变量的存储机制。Session 比 Cookie 更有优势：Session 是存储在服务器端的，不易被伪造；Session 的存储没有长度限制；Session 的控制更容易等。

PHP 主要是通过会话（Session）处理函数来对 Session 进行控制和使用。常用的处理函数如表 10.2 所示。

表 10.2 PHP 常用的会话处理函数

函　数	函 数 说 明
session_start();	开启 Session 或返回已经存在的 Session
$_SESSION['name'] = value;	注册一个 Session 变量
session_id()	设定或取得当前的 session_id 值
isset($_SESSION['name'])	检测指定的 Session 值是否存在。Isset()函数不只可以检测 Session，还可以检测其他类型，如 isset($_POST['name'])、isset($_GET['name'])等
session_regenerate_id()	更改 session_id 的值
session_name()	返回或改变当前 Session 的 name
unset($_SESSION['name'])	删除名为 name 的 Session
session_destroy()	结束当前会话，删除所有 Session

注意　（1）如果要改变当前 Session 的 name 值，必须在 session()之前调用 session_name()函数，而且 session_name 不能全部是数字，否则会不停地生成新的 session_id。

（2）不可以写成 unset($_SESSION)，这样会禁止整个会话的功能。





下面介绍几个使用 Session 时要注意的问题。

1．尽可能地将 session_start()放到第 1 行

这种情况是新手最容易犯的错误。产生的错误代码如下：

```
Warning: session_start() [function.session-start]: Cannot send session cache limiter - headers already sent…
```

其原因就是在使用 session_start()之前，有 HTML 代码输出。也许有的读者会说：session_start()之前没有任何代码，绝对没有，但是还是出现了这个错误。那么，请检查你的程序，是不是有空行，或类似 echo 语句的输出。如果有，请去掉，因为就算是一个小小的空格都是不可以的。所以为了避免这类错误的发生，尽可能地将 session_start()放到第 1 行。

2．在使用 Session 之前一定要先写 session_start()

大多数读者在使用 Session 之前都先调用 session_start()函数，但对于 session_destroy()函数却经常忽略。session_destroy()函数虽然是结束当前会话并删除所有 Session，但在删除之前，也要先开启 Session 支持才可以，否则会产生如下的错误代码：

```
session_destroy() [function.session-destroy]: Trying to destroy uninitialized session in…
```

所以，凡是在使用 Session 函数的页面中，都要加上 session_start()这句话。

3．删除所有 Session

如果想删除所有 Session，但又不想结束当前会话，用 unset 一个一个删除实在是太麻烦了，最简单的办法就是将一个空数组赋给$_SESSION，如$_SESSION = array()，这样就解决了。

10.7.4　冻结、解冻技术

所谓用户管理冻结技术，就是将用户冻结后，用户将无法登录，自然有冻就有解，如果解冻，用户便可以正常登录。用户冻结技术主要是针对一些不守网站规则的用户而建立的功能。其实用户冻结技术与用户推荐、文章审核等技术的实现原理是相同的。

采用数据库条件式方法，在指定数据库内设置单独的字段作为冻结技术的条件，当条件为真时用户正常运行解冻，当条件为假时用户停止运行冻结，这就是冻结和解冻的原理。本模块中具体的冻结、解冻操作是在 admin\member\member.php 文件中完成的，其代码如下：

```
$id = $_GET['id'];
$freeze = ($_GET['fz']+1) % 2;
$sql = "update tb_member set freeze = ".$freeze." where id = ".$id;
$cont = '冻结(解冻)操作，账号：'.$id;
$num = $conne->uidRst($sql);
if(!empty($num)){
$inssql = "insert into tb_log values('','".$cont."','".$_SESSION['manager']."',now())";
$num = $conne->uidRst($inssql);
```

10.7.5 配置文件

在配置文件中设置了服务器的目录、博客目录、上传图片目录、图片大小上限等信息。配置文件config.php 存储于前台。配置文件的代码如下：

```
<?php
define('PATH',$_SERVER['DOCUMENT_ROOT']);            //服务器目录
define('ROOT','/db_blog1.0 /');                      //博客目录
define('ADMIN','admin/');                            //后台目录
define('PIC','center/pics/image/');                  //上传图片目录
define('BAK','sqlbak/');                             //备份目录
define('HEADGIF','headgif/');                        //头像目录
define('MYSQLPATH','D:\\xampp\\MySQL\\bin\\');       //MySQL 执行文件路径
define('MYSQLHOST','127.0.0.1');                     //MySQL 服务器 IP
define('MYSQLDATA','db_blog');                       //MySQL 数据库
define('MYSQLUSER','root');                          //MySQL 账号
define('MYSQLPWD','111');                            //MySQL 密码
define('MAXSIZEPIC',500000);                         //允许上传的图片的最大字节权
?>
```

10.7.6 公共函数

博客系统的公共函数文件包括返回指定文件夹下的文件列表和判断上传文件的后缀。其存储于center\inc\func.php 中，函数代码如下：

```
<?php
//返回文件夹下的文件列表
function show_file($f_name){
$d_open = opendir($f_name);                          //打开指定目录
$num = 0;
while($file = readdir($d_open)){                     //循环输出目录下的文件
$filename[$num] = $file;                             //将文件存成数组
$num++;
}
closedir($d_open);                                   //关闭目录
return $filename;                                    //返回文件数组
}
/*
判断文件后缀
$f_type：允许文件的后缀类型
$f_upfiles：上传文件名
*/
function f_postfix($f_type,$f_upfiles){
```



```
$is_pass = false;
$tmp_upfiles = split("\.",$f_upfiles);                          //将后缀字符串分给数组
$tmp_num = count($tmp_upfiles);                                 //获取数组大小
if(in_array(strtolower($tmp_upfiles[$tmp_num - 1]),$f_type))   //上传的后缀是否在允许范围
$is_pass = $tmp_upfiles[$tmp_num - 1];                          //保存后缀
return $is_pass;                                                //返回结果
}
?>
```

10.8　本 章 小 结

本章的博客管理系统首先介绍了博客的基本概念、发展前景、影响范围及博客网的功能分类，使读者对当今主流博客有了一个大致的认识。其次，实现了一个博客系统，包含所有基本功能的项目开发，使读者对如何开发一个博客网站有了一个初步的了解。最后，希望读者通过自己的努力，来逐步完善和加强这个博客网的实用功能，最终达到一个令自己满意的作品。

第 11 章

创建 PDF 文档模块

（PHP+MySQL 实现）

PDF 文档格式是当前比较流行的电子文档与电子表格的一种标准格式。PDF 提供了完善的压缩处理，无论创建者创建的 PDF 文档使用何种字体、什么样的图片与版式设计，浏览者都可以通过免费的 Adobe Reader 对其进行阅读。本章开发一个支持 PDF 文档创建和下载功能的文章管理模块。

通过阅读本章，可以学习到：

- PHP 国际化
- FPDF 的安装与配置
- PDF 文档的创建
- 向 PDF 中插入图片
- 在 PDF 中绘制表格
- 在 PDF 中输出中文
- PDF 文档的下载

11.1　创建 PDF 文档模块概述

11.1.1　模块概述

创建 PDF 文档模块的重点是将用户添加的文章生成 PDF 文档，并且完成 PDF 文档的下载操作。其中还实现了文章添加、文章浏览等辅助功能，并且实现网站的国际化。既支持中文操作，同时也支持英文输出。

11.1.2　功能结构

创建 PDF 文档模块的功能结构如图 11.1 所示。

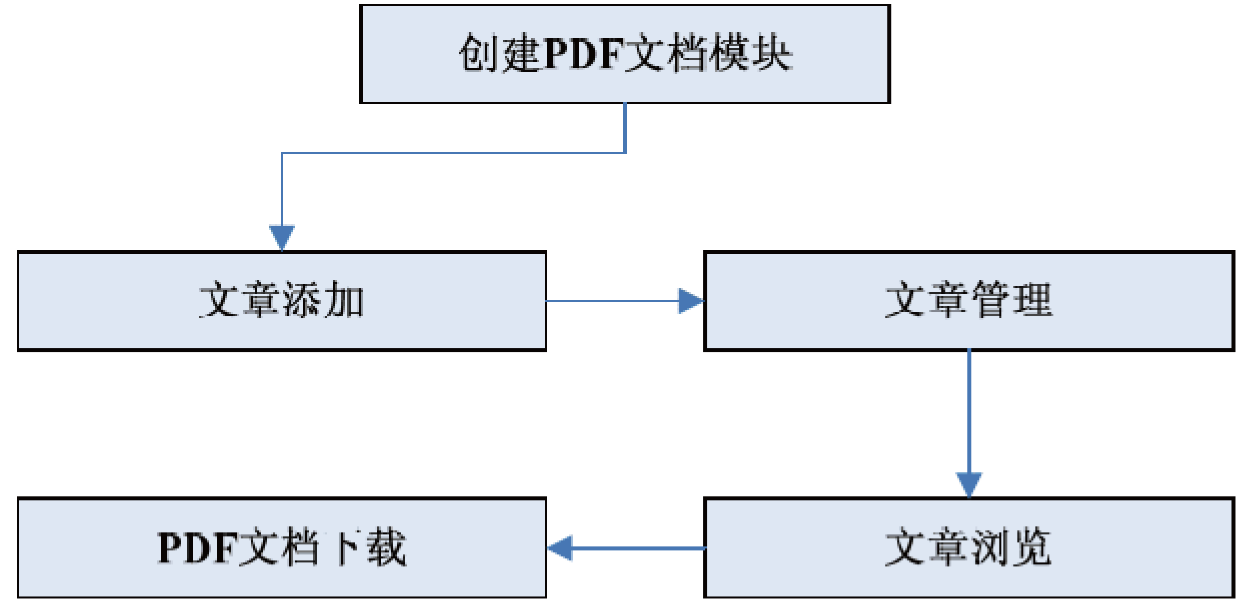


图 11.1　PDF 文档模块的功能结构图

11.1.3　程序预览

为了让读者更好地了解创建 PDF 文档模块的功能，下面列出创建 PDF 文档模块中一些功能的运行效果，供广大读者参考。

文章添加页面的运行效果如图 11.2 所示。



图 11.2　文章添加页面的运行效果

文章浏览运行效果如图 11.3 所示。



图 11.3　文章浏览运行效果

文章内容浏览、PDF 文档创建和下载链接的运行效果如图 11.4 所示。



图 11.4　文章内容浏览、PDF 文档创建和下载链接的运行效果

PDF 文档下载的运行效果如图 11.5 所示。

通过 Adobe Reader 阅读 PDF 文档的内容的运行效果如图 11.6 所示。

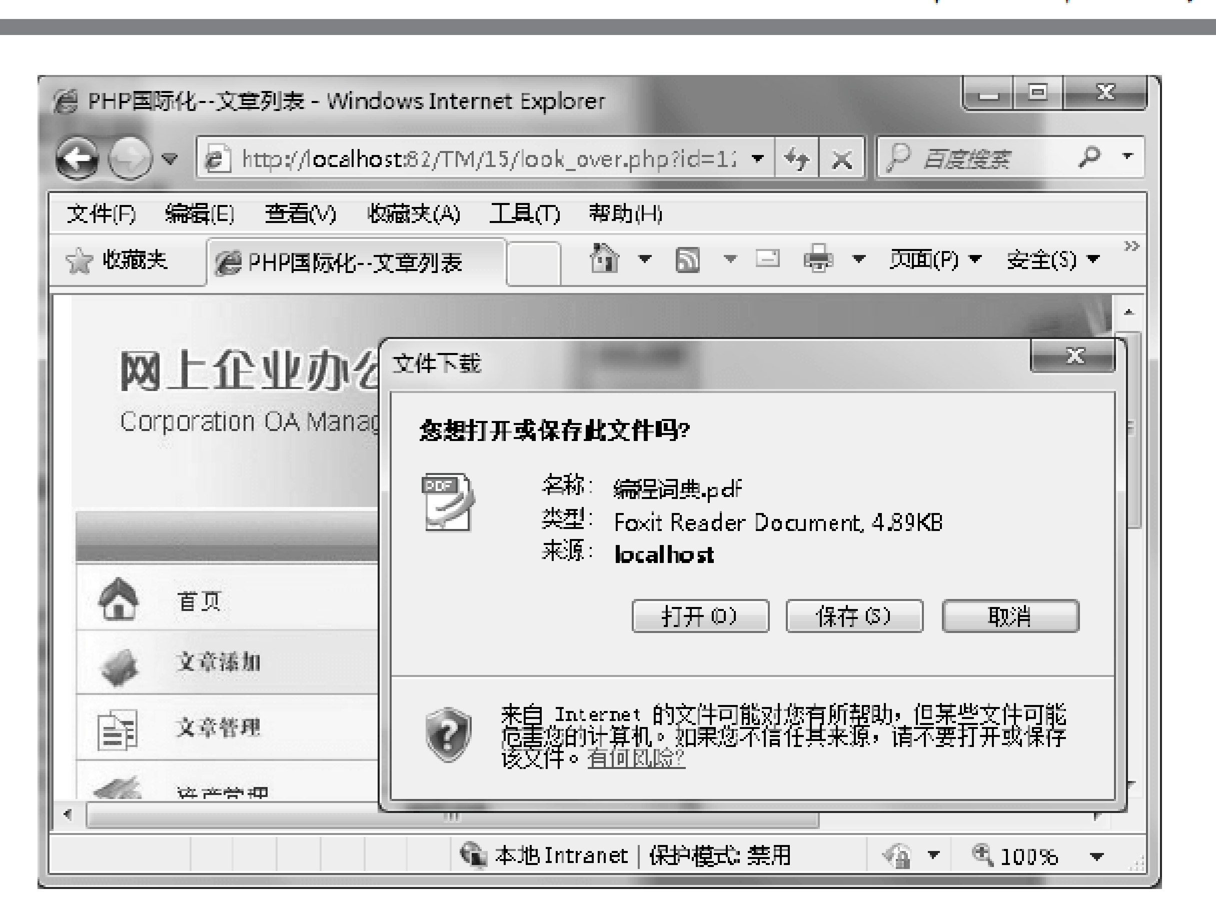


图 11.5　PDF 文档下载的运行效果

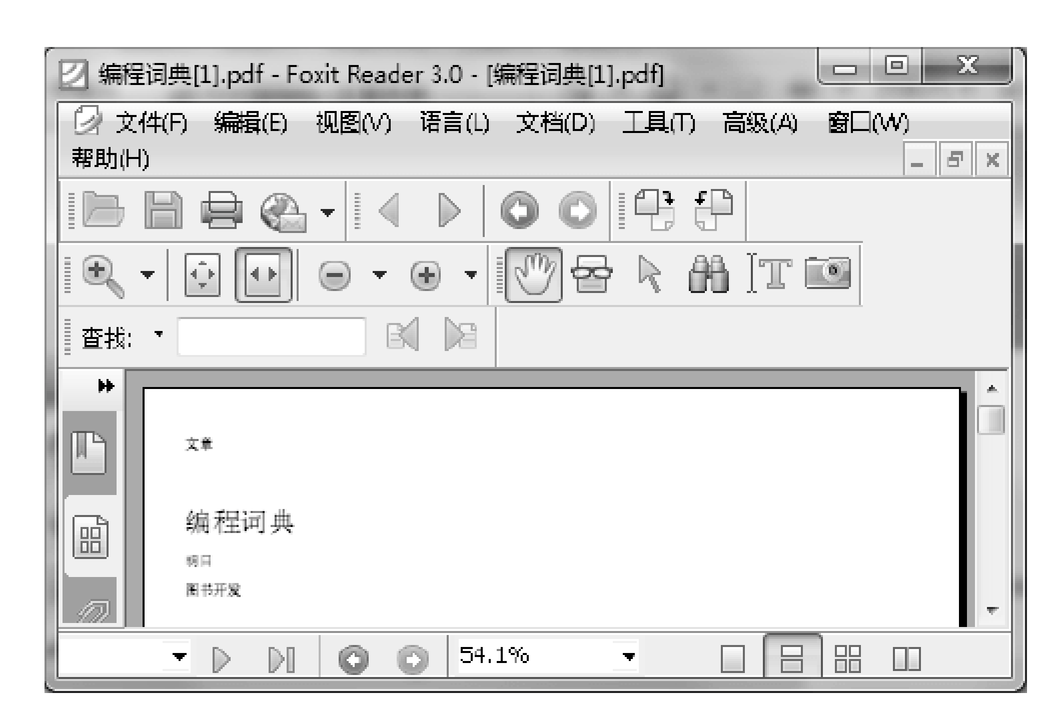


图 11.6　浏览 PDF 文档的内容

11.2　数据库设计

在本模块中通过 MySQL 数据库存储用户上传的文章，然后通过 PDF 格式将文章内容下载到本地。

11.2.1　创建数据库

在本模块中，创建 db_pdf 数据库存储用户上传的文章数据。db_pdf 数据库中只包含一个数据表，即 tb_articles。数据库的结构及数据表的情况如图 11.7 所示。



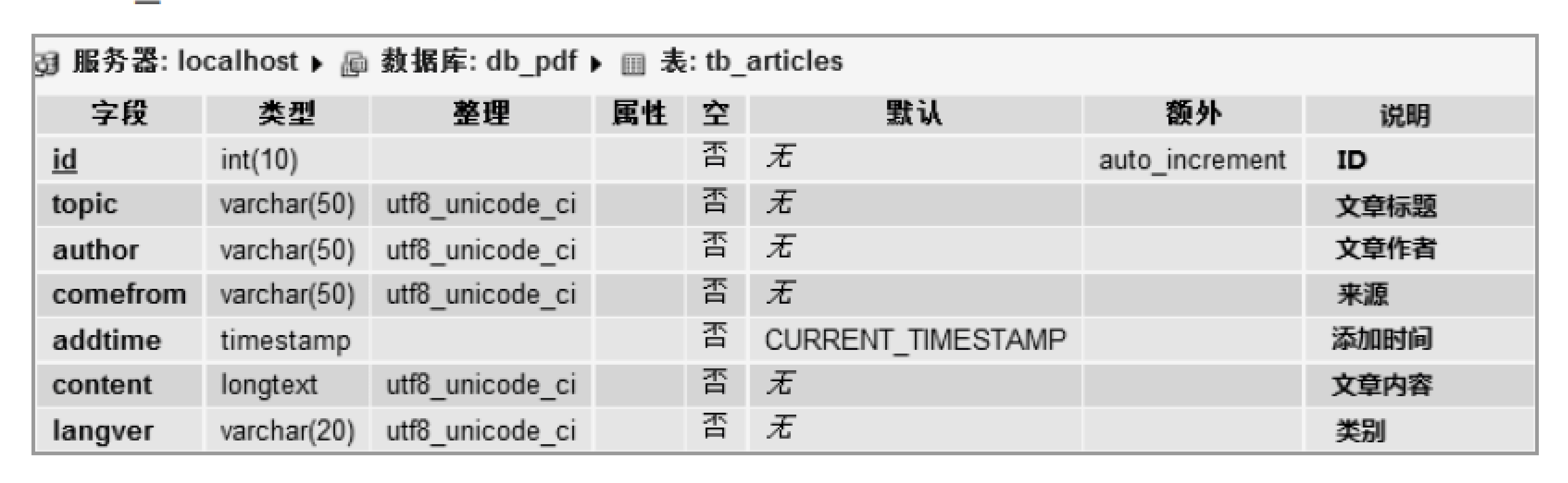
服务器: localhost ▸ 数据库: db_pdf

表	操作	记录数	类型	整理
tb_articles		4	MyISAM	utf8_unicode_ci
1 个表	总计	4	MyISAM	utf8_general_ci

图 11.7　db_pdf 数据库结构及数据表的情况

11.2.2　创建数据表

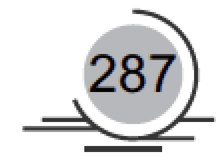
在 tb_articles 数据表中存储用户提交的文章信息，包括 id、文章标题、文章作者、来源、添加时间、文章内容和类别。tb_articles 数据表的结构如图 11.8 所示。



服务器: localhost ▸ 数据库: db_pdf ▸ 表: tb_articles

字段	类型	整理	属性	空	默认	额外	说明
id	int(10)			否	无	auto_increment	ID
topic	varchar(50)	utf8_unicode_ci		否	无		文章标题
author	varchar(50)	utf8_unicode_ci		否	无		文章作者
comefrom	varchar(50)	utf8_unicode_ci		否	无		来源
addtime	timestamp			否	CURRENT_TIMESTAMP		添加时间
content	longtext	utf8_unicode_ci		否	无		文章内容
langver	varchar(20)	utf8_unicode_ci		否	无		类别

图 11.8　tb_articles 数据表的结构



11.2.3　连接数据库

本模块中将连接数据库的操作存储于 Connections\conn.php 文件中，设置数据库的用户名是 root，密码是 111。conn.php 文件的代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die('连接失败:' . mysql_error());     //连接服务器
mysql_select_db("db_pdf",$conn) or die ('数据库选择失败:' . mysql_error());           //选择数据库
mysql_query("set names utf8");                                                    //设置页面编码格式
?>
```

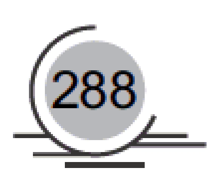
11.3　模块国际化设置

本模块的国际化功能只是一个辅助，它让文章的添加和浏览功能支持中文和英文两种语言。其中，模块国际化设置的步骤如下：

（1）创建 lang 文件夹，编写 lang_ch.php 和 lang_en.php 文件，分别定义中文语言配置文件和英文语言配置文件，代码请参考光盘中的内容。

（2）创建 config.php 文件，获取浏览器中语言选项的值，根据语言选项的值调用不同的语言配置文件。代码如下：

```
<?php
session_start();
$lang=$_SERVER['HTTP_ACCEPT_LANGUAGE'];          //获取语言选项
$lang=explode(';',$lang);                        //以“;”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
$lang=explode(',',$lang);                        //以“,”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
switch($lang){                                   //根据获取到的语言选项的值，为变量赋值
case "en-us":
$lang="en";
break;
case "zh-cn":
$lang="ch";
break;
default :
$lang="en";
break;
}
if(isset($_SESSION['lang'])){
$lang=$_SESSION['lang'];                         //为 Session 变量赋值
}
include "lang/lang_$lang.php";                   //调用不同的语言配置文件
?>
```



（3）创建 setlang.php 文件，根据超链接中传递的参数，为 Session 变量赋值，并且调用不同的操作页面。代码如下：

```
<?php
session_start();
switch($_GET['lang']){                    //获取超链接传递的语言选项的值
case "en":
$_SESSION['lang']="en";                   //根据超链接的值，为 Session 变量赋值
break;
case "ch":
$_SESSION['lang']="ch";
break;
default :
$_SESSION['lang']="en";
break;
}
if($_GET['link']=="addnew"){              //根据超链接的值，跳转到不同操作页面
header("Location:add_data.php");
}elseif ($_GET['link']=="show"){
header("Location:look_over.php");
}else{
header("Location:index.php");
}
?>
```

至此，PHP 国际化设置完毕，下面讲解模块具体功能的开发。

11.4　文 章 浏 览

11.4.1　文章浏览概述

文章浏览功能默认输出英文格式的文章标题数据，运行结果如图 11.9 所示。当单击文章标题的超链接后，将浏览到文章的详细信息，并且在详细信息页还设置了 PDF 文档下载的超链接，运行效果如图 11.10 所示。

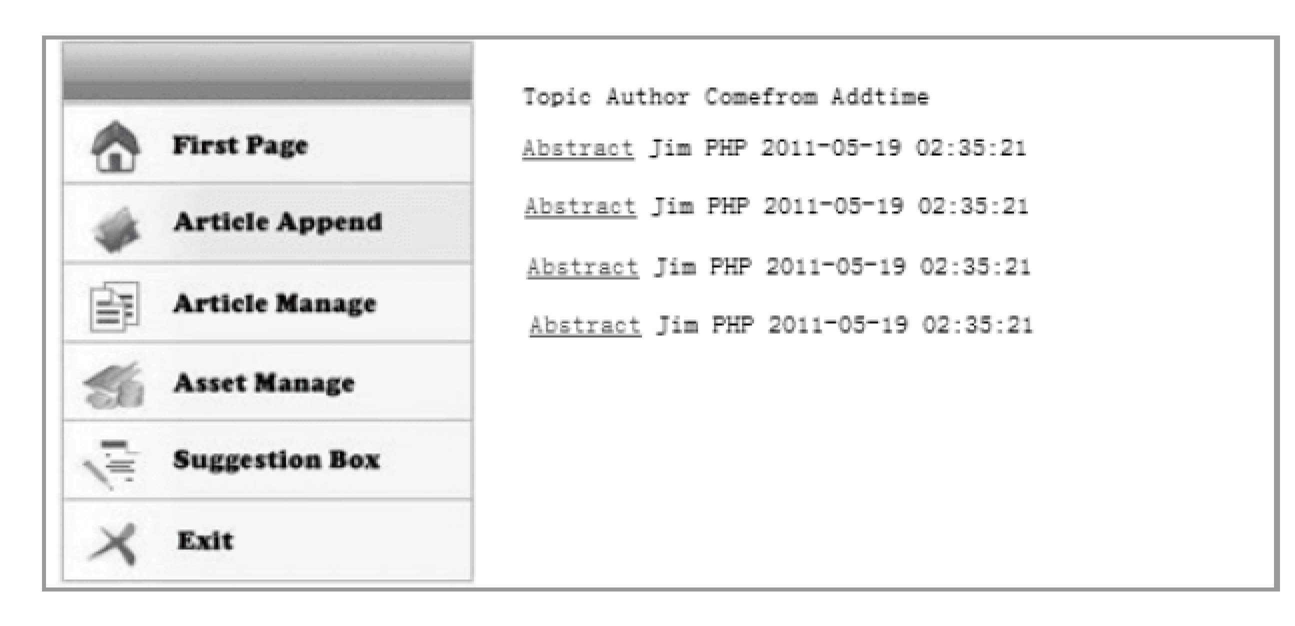


图 11.9　英文格式的文章列表



图 11.10　英文格式的文章浏览

11.4.2　文章列表实现过程

创建 index.php 文件，根据选择的语言不同，输出不同的页面和内容。首先，通过 require_once()语句调用不同的包含文件，完成数据库连接和语言配置；然后，执行查询操作，从数据表中读取出指定的数据信息；最后，在页面中输出数据。完整代码如下：

```
<?php
require_once('Connections/conn.php');                                          //包含数据库连接文件
require_once('config.php');                                                    //包含语言配置文件
$query_rs="select * from tb_articles where langver='$rs_filter' order by id desc";  //定义查询语句
$rs=mysql_query($query_rs,$conn) or die(mysql_error());                        //执行查询语句
$row_rs=mysql_fetch_assoc($rs);                                                //获取查询结果
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><?php echo $index_title ?></title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body class="twoColLiqLtHdr">
<div id="container">
<div id="header"><img src="images/<?php if($rs_filter=="en"){echo "OA_021.jpg";}else{echo "OA_02.jpg";}?>"
width="780" height="120" border="0" usemap="#Map" />
<map name="Map" id="Map"><area shape="rect" coords="709,73,776,95" href="setlang.php?lang=ch&link=
addnew"/>
<area shape="rect" coords="706,98,772,113" href="setlang.php?lang=en&link=addnew" /></map>
<!-- end #header --></div>
<div id="sidebar1">
<img src="images/<?php if($rs_filter=="en"){echo "OA_061.jpg";}else{echo "OA_06.jpg";}?>"
```



```
width="188" height="396" border="0" usemap="#Map2" />
<map name="Map2" id="Map2"><area shape="rect" coords="47,110,154,131" href="index.php" />
<area shape="rect" coords="43,71,163,98" href="add_data.php" /></map>
<!-- end #sidebar1 --></div>
<div id="mainContent">
<P><?php echo $index_topic; ?>
<?php echo $index_author; ?>
<?php echo $index_comefrom; ?>
<?php echo $index_addtime; ?>
</P>
<?php
do{
?><P>
<a href="look_over.php?id=<?php echo $row_rs['id'];?>"><?php echo $row_rs['topic'];?></a>
<?php echo $row_rs['author'];?>
<?php echo $row_rs['comefrom'];?>
<?php echo $row_rs['addtime'];?>
</P>
<?php
}while($row_rs=mysql_fetch_assoc($rs));
mysql_free_result($rs);
?>
<!-- end #mainContent --></div>
<div id="footer"><img src="images/OA_13-14.jpg" width="780" height="78" />
<!-- end #footer --></div>
<!-- end #container --></div>
</body>
</html>
```

注意　index.php 文件中输出的内容，会根据中文和英文不同语言传递的参数值进行切换输出。

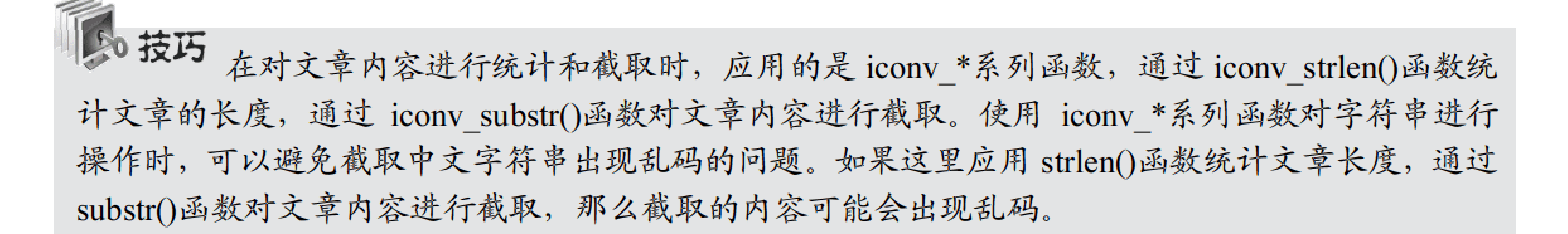
11.4.3　文章浏览实现过程

当用户单击文章的标题时，跳转到 look_over.php 文件中，在该文件中根据超链接传递的文章 id 查询出该文章的详细信息，并输出文章的内容。同时在 look_over.php 文件中创建“下载 PDF 文档”超链接，链接到 setup_pdf.php，在该文件中完成 PDF 文档的生成和下载操作。look_over.php 文件的关键代码如下：

```
<?php
require_once('Connections/conn.php');                                    //连接数据库
require_once('config.php');                                              //配置语言
$colname_rs='1';                                                         //定义变量值
if(isset($_GET['id'])){                                                  //检测变量是否被设置
$colname_rs=(get_magic_quotes_gpc())?$_GET['id']:addslashes($_GET['id']);   //对数据格式进行转换
}
$query_rs=sprintf("select * from tb_articles where id=%s",$colname_rs);      //对字符串进行格式化
```

```
$rs=mysql_query($query_rs,$conn) or die(mysql_error());                    //执行查询操作
$row_rs=mysql_fetch_array($rs);                                            //获取查询结果集
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><?php echo $index_title ?></title>
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body class="twoColLiqLtHdr">
<div id="container">
<div id="header"><img src="images/<?php if($rs_filter=="en"){echo "OA_021.jpg";}else{echo "OA_02.jpg";}?>"
width="780" height="120" border="0" usemap="#Map" />
<map  name="Map"  id="Map"><area  shape="rect"  coords="708,73,775,95"  href="setlang.php?lang=ch&link=
addnew" />
<area shape="rect" coords="706,98,772,113" href="setlang.php?lang=en&link=addnew" /></map>
<!-- end #header --></div>
<div id="sidebar1">
<img src="images/<?php if($rs_filter=="en"){echo "OA_061.jpg";}else{echo "OA_06.jpg";}?>"
width="188" height="396" border="0" usemap="#Map2" />
<map name="Map2" id="Map2"><area shape="rect" coords="47,111,154,132" href="index.php" />
<area shape="rect" coords="43,71,163,98" href="add_data.php" /></map>
<!-- end #sidebar1 --></div>
<div id="mainContent">
<P><?php echo $row_rs['topic'];?></p>
<P><?php echo $show_author; ?>：<?php echo $row_rs['author'];?></p>
<P><?php echo $show_comefrom;?>：<?php echo $row_rs['comefrom'];?></p>
<P><?php
$len=iconv_strlen($row_rs['content'],'utf-8');                       //对文章内容长度进行统计
if($len>100){
echo iconv_substr($row_rs['content'],0,100,'utf-8')."......";        //对文章内容进行截取输出
}else{
echo $row_rs['content'];                                             //输出文章内容
}
?></p>
<P><?php echo $show_upttime;?>：<?php echo $row_rs['addtime'];?></p>
<P><a href="setup_pdf.php?id=<?php echo $row_rs['id']; ?>"
target="_blank"><?php echo $show_down;?></a></p>
<!-- end #mainContent --></div>
<div id="footer"><img src="images/OA_13-14.jpg" width="780" height="78" />
<!-- end #footer --></div>
<!-- end #container --></div>
</body>
</html>
<?php
mysql_free_result($rs);
?>
```





技巧　在对文章内容进行统计和截取时，应用的是 iconv_*系列函数，通过 iconv_strlen()函数统计文章的长度，通过 iconv_substr()函数对文章内容进行截取。使用 iconv_*系列函数对字符串进行操作时，可以避免截取中文字符串出现乱码的问题。如果这里应用 strlen()函数统计文章长度，通过 substr()函数对文章内容进行截取，那么截取的内容可能会出现乱码。

11.4.4　PDF 文档创建和下载

PDF 文档的下载应用的是超链接下载方式，而 PDF 文档的创建在 setup_pdf.php 文件中完成，其根据超链接传递的 id 值，从数据库中查询出指定文章的详细信息，应用 PDF_Chinese 类中的方法完成 PDF 文档的创建。setup_pdf.php 文件的完整代码如下：

```
<?php
header("Content-Type:text/html;charset=gb2312");                //设置页码编码格式
include('pdf/chinese.php');                                     //载入中文组件
include('Connections/conn.php');                                //载入数据库连接文件
class PDF extends PDF_Chinese{                                  //继承中文类
function Header(){                                              //定义方法设置标题
$this->SetFont('gb','',10);                                     //设置字体
$this->Write(10,'文章');                                        //输出标题内容
$this->Ln(20);                                                  //换行
}
function Footer(){                                              //设置页脚
$this->SetY(-15);
$this->SetFont('gb','',10);
$this->Cell(0,10,'第'.$this->PageNo().'页');
}
}
$colname_rs_id=$_GET['id'];                                     //获取指定数据的 id
$query_rs_article=sprintf("select * from tb_articles where id=%s",$colname_rs_id);  //定义 SQL 语句
$rs_article=mysql_query($query_rs_article,$conn) or die(mysql_error());             //执行 SQL 查询语句
$row_rs_article=mysql_fetch_assoc($rs_article);                 //获取查询结果
$pdf=new PDF();                                                 //实例化 PDF 类
$pdf->AddGBFont();                                              //设置字体
$pdf->Open();                                                   //打开文件
$pdf->AliasNbPages();                                           //为每页定义一个别名
$pdf->AddPage();                                                //分页方法
$pdf->SetFont('gb','B',20);                                     //设置字体
$pdf->Cell(0,10,iconv("utf-8","gb2312",$row_rs_article['topic']));//输出一个内容
$pdf->Ln();                                                     //换行
$pdf->SetFont('gb','',10);
$pdf->Cell(0,10,iconv("utf-8","gb2312",$row_rs_article['author']));
$pdf->Ln();
$pdf->SetFont('gb','',10);
$content= iconv('utf-8','gb2312',$row_rs_article['content']);   //设置编码格式
$pdf->MultiCell(0,5,$content);                                  //输出内容，并且执行终止
```

```
$pdf->Ln();
$pdf->AddPage();                                                      //分页
$pdf->Output(iconv("utf-8","gb2312",$row_rs_article['topic']).'.pdf',true);   //生成 PDF 文件
?>
```

说明 有关通过 FPDF 创建 PDF 文档的详细讲解请参考本章的 11.6 节。

11.5 文 章 添 加

11.5.1 文章添加概述

文章添加功能将用户提交的文章数据添加到指定的数据表中。英文格式文章添加的运行效果如图 11.11 所示。

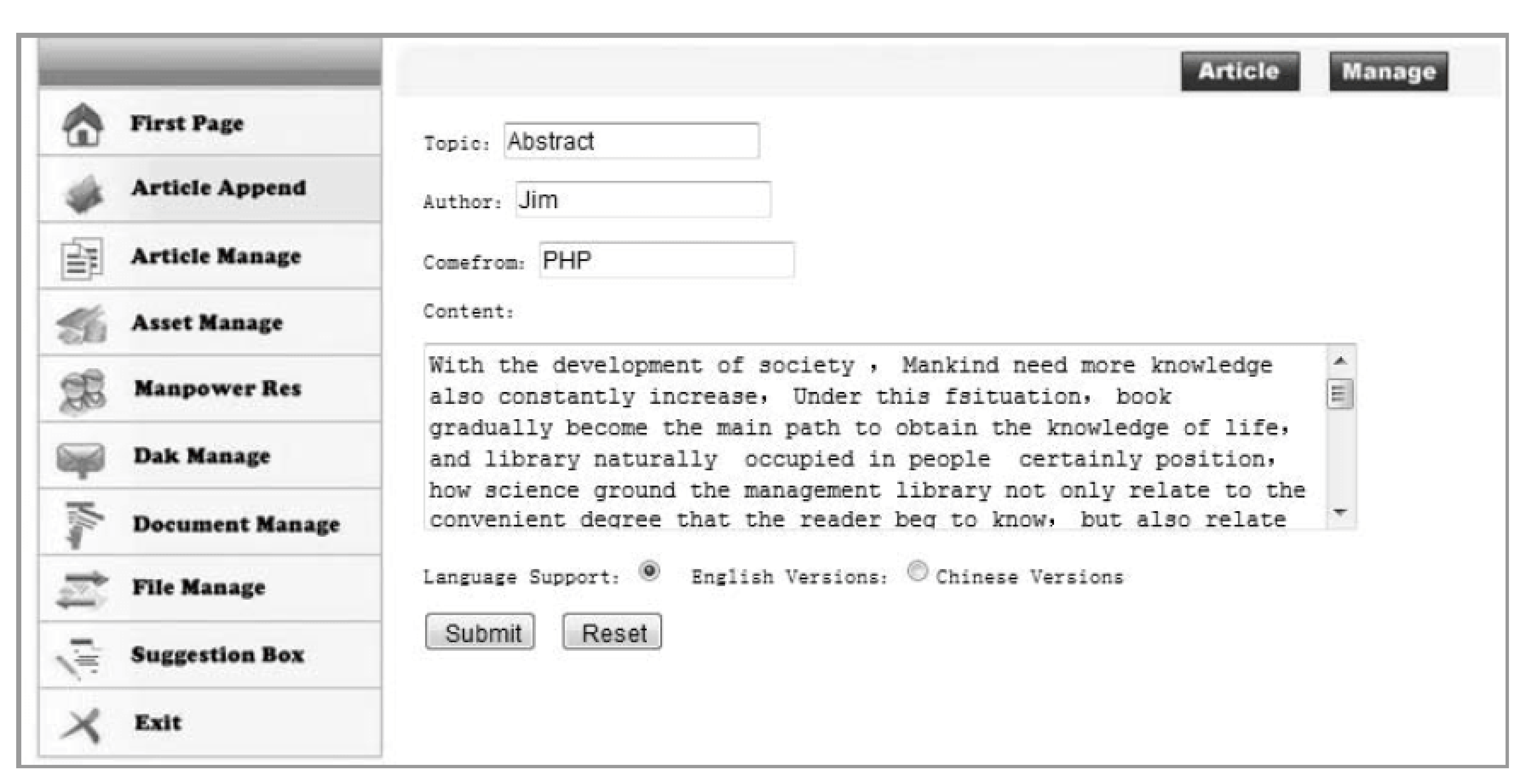


图 11.11 英文格式文章添加的运行效果

11.5.2 文章添加实现过程

文章添加操作在 add_data.php 文件中完成。可以将其划分为两部分：第一部分是通过表单提交文章数据；第二部分是获取表单提交的数据，将其添加到指定的数据表中。

第一部分创建表单的代码如下：

```
<form name="form" method="post" action="">
<p><?php echo $add_data_topic; ?>：<input type="text" name="topic" value="" /></p>
<p><?php echo $add_data_author; ?>：<input type="text" name="author" value="" /></p>
<p><?php echo $add_data_comefrom; ?>：<input type="text" name="comefrom" value="" /></p>
<p><?php echo $add_data_content; ?>：  </p>
<textarea name="content" id="textarea" cols="65" rows="6"></textarea>
<p><?php echo $add_data_langsp; ?>：
```



```
<input type="radio" name="langver" value="en" />  <?php echo $add_data_lenver ?>:
<input type="radio" name="langver" value="ch" /><?php echo $add_data_lchver ?></p>
<p><input type="submit" name="Submit" value="<?php echo $add_data_submit ?>" />  
<input type="reset" name="Reset" value="<?php echo $add_data_reset ?>" /></p>
</form>
```

第二部分的代码如下：

```
<?php
header("Content-type:text/html;charset=utf-8");        //设置页面编码格式
require_once('Connections/conn.php');                  //连接数据库
require_once('config.php');                            //配置语言
//定义格式转换的方法
function GetSQLValueString($theValue,$theType,$theDefinedValue="",$theNotDefinedValue=""){
$theValue=(!get_magic_quotes_gpc()) ? addslashes($theValue):$theValue;
switch($theType){
case "text";
$theValue=($theValue !="")? "'". $theValue. "'" : "NULL";
break;
}
return $theValue;
}
if(isset($_POST['Submit'])){
//获取表单中提交的数据，并完成对数据格式的转换
$insertSQL=sprintf("insert into tb_articles(topic,author,comefrom,content,langver) values(%s,%s,%s,%s,%s)",
GetSQLValueString($_POST['topic'],"text"),
GetSQLValueString($_POST['author'],"text"),
GetSQLValueString($_POST['comefrom'],"text"),
GetSQLValueString($_POST['content'],"text"),
GetSQLValueString($_POST['langver'],"text")
);
//执行添加语句，将数据添加到数据表中
$Result1=mysql_query($insertSQL,$conn) or die(mysql_error());
echo "<script>alert('$add_data_Suc'); window.location.href='add_data.php';</script>";
}
?>
```

说明 本模块中的功能偏少。因为本模块的重点是讲解 PDF 文档的创建，所以这里只选择文章浏览和文章添加作为辅助内容。如果读者感兴趣，可以在本模块基础上进行二次开发，增加更多更好的功能。

11.6　技 术 提 炼

在 11.4.4 节中讲解的 PDF 文档的创建和下载，只是从整个模块的角度出发，简单地给出了 PDF 文档创建和下载所用的文件，以及程序代码。下面将对 FPDF 类中的方法进行系统讲解。

11.6.1 FPDF 概述

FPDF 是一个免费的 PDF 操作类，通过它可以完成基本的 PDF 创建功能，并且它还支持中文（需要对相应字体进行配置）。FPDF 类中常用的 PDF 操作函数包括 Open()函数、AddPage()函数，SetFont()函数、Cell()函数和 Output()函数。

FPDF 的下载可以通过访问 http://www.fpdf.org/来得到，将下载的文件直接解压到某一文件夹即可。

11.6.2 创建简单的 PDF 文档

通过 FPDF 创建 PDF 文档，首先要实例化 FPDF 类，然后调用类中的函数（Open()函数、AddPage()函数、SetFont()函数、Cell()函数和 Output()函数）完成文档的创建。

（1）创建 FPDF 对象的语法格式如下：

```
$pdf = new FPDF([string page-orientation [, string measure-unit [, string page-format]]])
```

FPDF 对象的参数说明如表 11.1 所示。

表 11.1 FPDF 对象的参数说明

参　　数	说　　明
page-orientation	设置 PDF 文档的排版。可选值为：P 表示竖向；L 表示横向
measure-unit	设置文档中位置的计量单位。可选值为：pt（点）、mm（毫米）、cm（厘米）、in（英寸）
page-format	设置 PDF 文档纸张的类型。可选值为：表示纸张类型的字符串，例如“A4”、“A5”、“Letter”；或者一个包含有两个元素的二维数组，直接指明纸张的大小

（2）Open()函数用于标识开始创建 PDF 文档，其语法格式如下：

```
$pdf->Open()
```

（3）AddPage()函数为 PDF 文档添加一个新页，其语法格式如下：

```
$pdf ->AddPage([string page-orientation])
```

其中，page-orientation 表示创建的 PDF 文档是横向还是竖向的。与前面介绍的创建对象时的使用方法相同。

（4）SetFont()函数设置当前使用的字体，其语法格式如下：

```
$pdf->SetFont(string font [, string style [, float size]]);
```

其中，font 表示字体。style 表示样式，style 的可选值有如下 3 种。如果没有指定 style 的值，则默认为普通的形式。

☑ B：粗体。
☑ I：斜体。
☑ U：下滑线。

size表示字体的大小。如果不指定则取默认值，为12pt。





（5）Cell()函数用于为当前 PDF 文档增加一个单元格。其语法格式如下：

```
$pdf->Cell(float w [, float h [, string txt [, mixed border [, int ln [, string align [, int fill [, mixed link]]]]]]])
```

Cell()函数的参数说明如表 11.2 所示。

表 11.2　Cell()函数的参数说明

参　数	说　明
w	表示单元格宽度
h	表示单元格高度
txt	表示单元格中添加的文本字符串
border	表示单元格的边框。设置为 0 表示无边框；设置为 1 表示边框。或者 L 表示左边框；T 表示顶部边框；R 表示右边框，B 表示底部边框
ln	表示换行后，设置目前的位置。可选值为：0 表示向右移动；1 表示定位到下一行的开始；2 表示向下移动。默认值为 0
align	设置字符串的对齐方式。可选值为：L 表示左对齐；C 表示居中；R 表示右对齐。默认值为 L
fill	设置单元格的背景。可选值为：0 表示透明；1 表示不透明。默认值为 0
link	设置一个 URL 链接或者由 AddLink()函数添加的内部链接

（6）Output()函数用于输出 PDF 文档，其语法格式如下：

```
$pdf->Output([string filename [, bool download]])
```

其中，filename 表示要存储的文件名。如果不指定文件名，则浏览时会在 IE 浏览器中直接打开。download 表示是否一定要用户下载查看，默认为 false。

下面应用上述介绍的函数，创建一个 PDF 文档，输出一段英文字符串。example1.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');          //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');             //包含 FPDF 类库文件
$pdf=new FPDF('P', 'mm', 'A4');           //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小为 A4
$pdf->Open();                             //开始创建 PDF
$pdf->AddPage();                          //增加一页
$pdf->SetFont('Courier','I',20);          //设置字体样式
$pdf->Cell(0,0,'mrsoft!');                //增加一个单元格
$pdf->Output();                           //输出 PDF 到浏览器
?>
```

运行结果如图 11.12 所示。

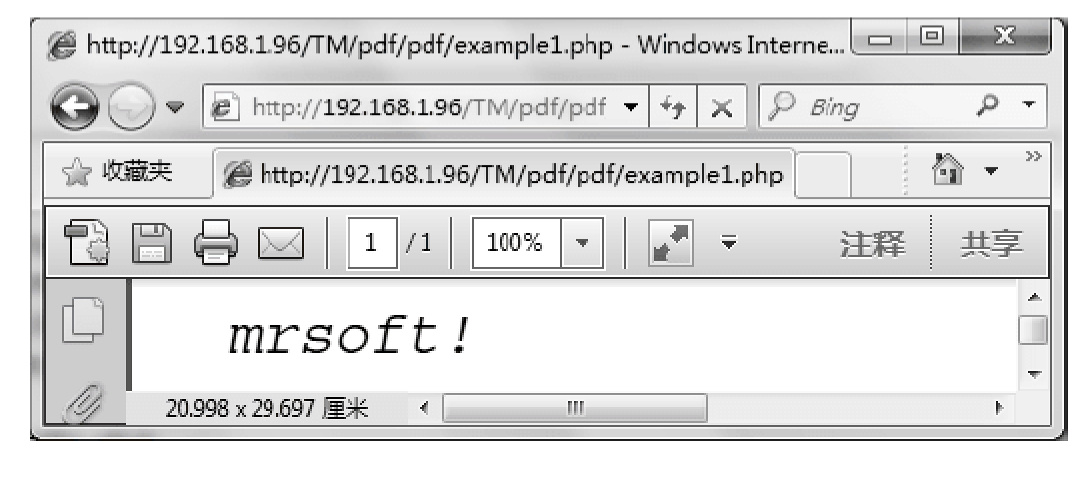


图 11.12　简单的 PDF 文档

技巧 上面的示例是将 PDF 文档的内容在 IE 浏览器中输出，您可以将上述代码中的“$pdf->Output();”修改为“$pdf->Output('mrsoft.pdf',true);”，表示定义 PDF 文档名称为 mrsoft.pdf，设置用户下载查看。这样，就完成了将 PDF 文档保存到服务器并提示用户下载的操作。

11.6.3　在 PDF 中插入图片

在 FPDF 中通过 Image()函数向 PDF 中插入图片，其语法格式如下：

```
Image(string file, float x, float y, float width, float height)
```

其中，x 和 y 表示图片所在的坐标；width 和 height 表示图片的宽度和高度。

下面应用 Image()函数向 PDF 文档中插入一张图片，并且将 PDF 在 IE 浏览器中直接输出。example2.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');                //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                   //包含 FPDF 类库文件
$pdf=new FPDF('P', 'mm', 'A4');                 //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小为 A4
$pdf->Open();                                   //创建 PDF
$pdf->AddPage();                                //增加一页
$pdf->SetFont('Courier','I',20);                //设置字体样式
$pdf->Image('images/as.jpg',10,10,80,80);       //插入图片
$pdf->Output();                                 //输出 PDF 到浏览器
?>
```

其运行效果如图 11.13 所示。

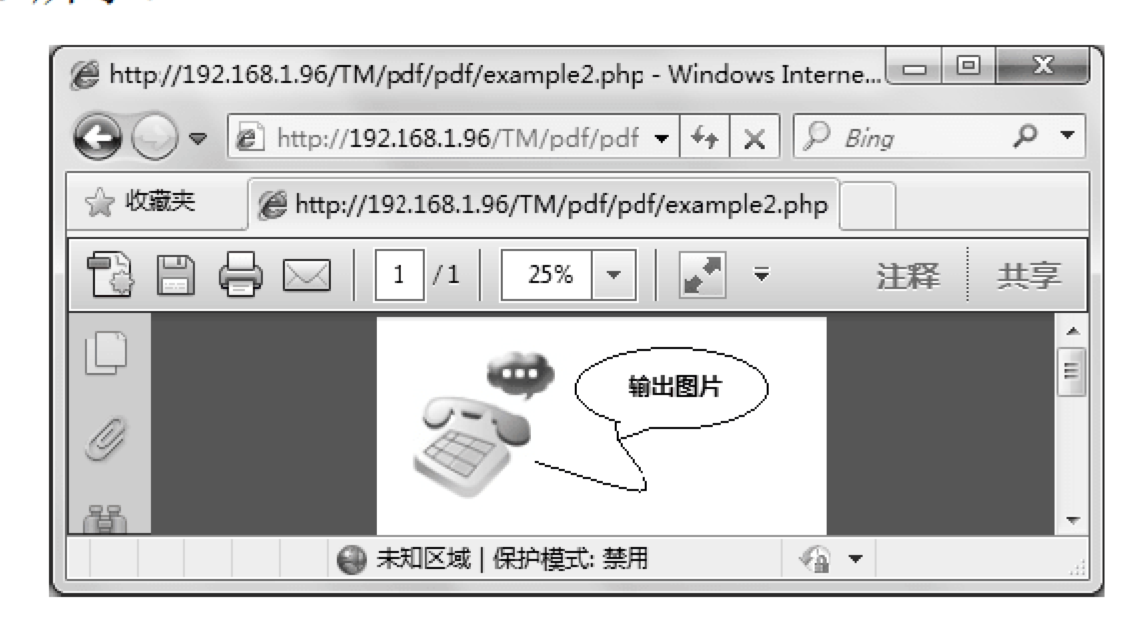


图 11.13　在 PDF 中插入图片

11.6.4　设置 PDF 文档的页眉与页脚

在 FPDF 中，页眉与页脚是通过重写 FPDF 类中的 Header()方法和 Footer()方法来实现的。因为 FPDF 类中的 Header()方法和 Footer()方法虽然存在，但是方法体中没有任何内容。所以，设置 PDF 文档的页眉与页脚需要重写 FPDF 类中的 Header()方法和 Footer()方法，完成页眉与页脚的设置。

下面创建 PDF 类，继承 FPDF 类，在 PDF 类中对 Header 方法和 Footer 方法进行重写，完成页眉





与页脚的设置。然后创建 PDF 文档，输出一行英文字符串和一张图片。example3.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');                //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                   //包含 FPDF 类库文件
class PDF extends FPDF{                         //继承 FPDF 类，重写页眉与页脚的方法
function Header(){                              //设置页眉
$this->SetFont('Arial','B',15);                 //设置页眉字体
$this->Write(10,'Title');                       //写入页眉文字
$this->Ln(20);                                  //换行
}
function Footer(){                              //设置页脚
$this->SetY(-15);                               //设置页脚所在位置
$this->SetFont('Arial','I',8);                  //设置页脚字体
$this->Cell(0,10,'Page - '.$this->PageNo());    //输出当前页码作为页脚内容
}
}
$pdf=new PDF('P', 'mm', 'A4');                  //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小为 A4
$pdf->Open();                                   //创建 PDF
$pdf->AddPage();                                //增加一页
$pdf->SetFont('Courier','I',20);                //设置字体样式
$pdf->Cell(30,30,'MRSOFT!');                    //输出字符串
$pdf->Image('images/as.jpg',30,50,80,80);       //插入图片
$pdf->Output();                                 //输出 PDF 到浏览器
?>
```

其运行结果如图 11.14 所示。

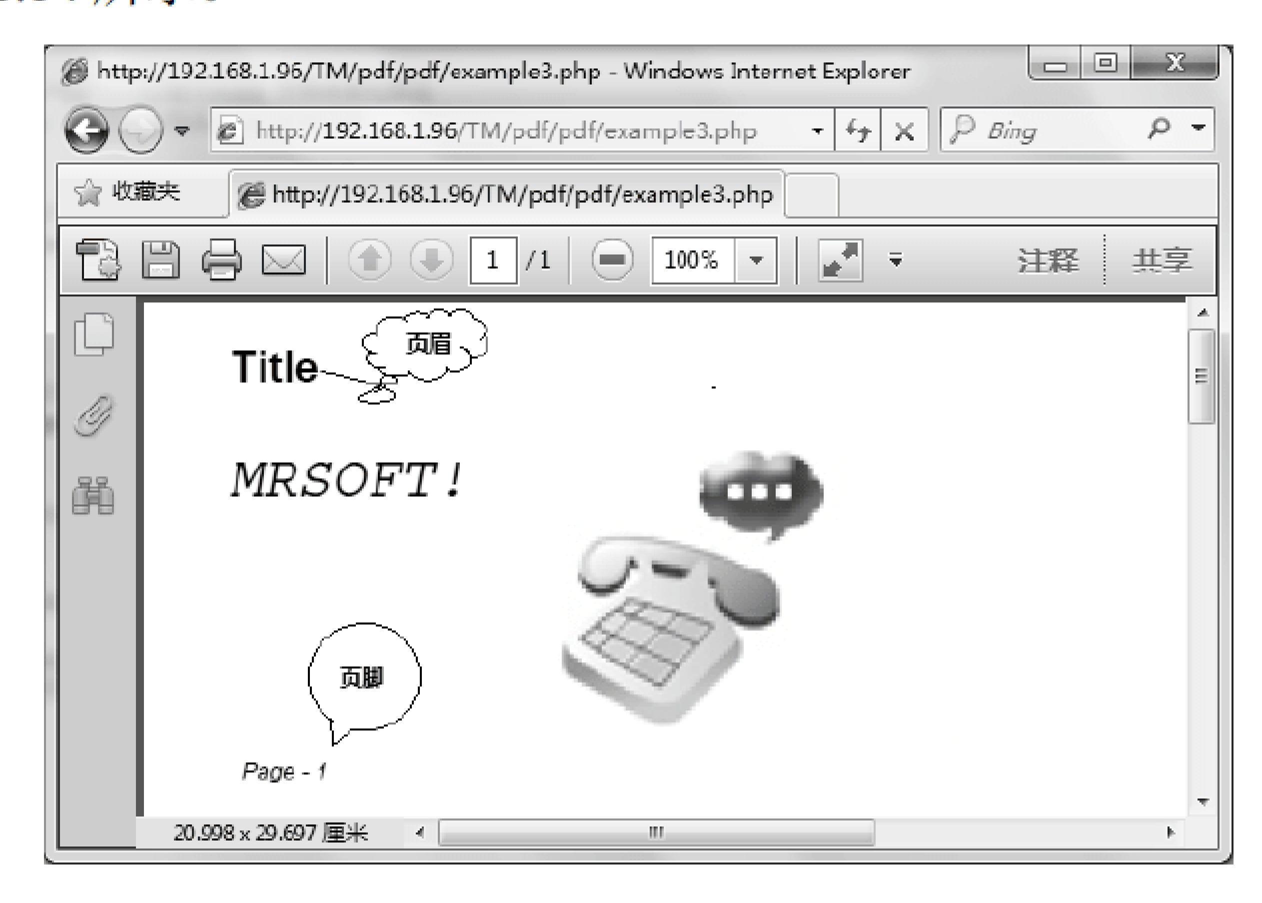


图 11.14　设置 PDF 的页眉和页脚

在设置 PDF 的页眉与页脚时，引入了 Write()、Ln()、PageNo()和 SetY() 4 个方法。

Write()方法用于输出字符串。当到达文档的右边位置或者遇到（\n）时，会自动换行。其语法格式如下：

```
Write(float h, string txt [, mixed link])
```

参数h定义字符串的行列高度；参数txt指定输出的字符串；参数link设置链接的网页或者AddLink()方法的标识符。

Ln()方法用于换行操作，其语法格式如下：

```
Ln([float h])
```

参数 h 用于设置行的高度。默认值为最后输出行的高度。

PageNo()方法的作用是返回当前页码，其语法格式如下：

```
int PageNo();
```

SetY()方法用于设置页脚的坐标。其语法格式如下：

```
SetY(float y);
```

参数 y 为页面上的 Y 坐标，单位为毫米。如果 y 为负数，则表示从页面底部向上的距离。例如，上面的示例中设置 y 的值为-15，即页脚所在位置距页面底部 15 毫米。

11.6.5 在 PDF 文档中绘制表格

通过 FPDF 绘制表格的方法与输出文字的方法类似，在输出文字时往往指定 Cell()方法的边框参数为 0，即不输出边框。在绘制表格时往往将其设置成大于 0 的整数，用于表示边框，这就是 PDF 文档中表格的单元格。当我们把这些单元格组合到一起，就构成了表格。

下面看一个输出表格的示例，example4.php 的代码如下：

```
<?php
define('FPDF_FONTPATH','font/');              //定义 font 文件夹所在路径
require_once('pdf/fpdf.php');                 //包含 FPDF 类库文件
class PDF extends FPDF{
function Header(){                            //设置页眉
$this->SetFont('Arial','B',15);               //设置页眉字体
$this->Write(10,'Pay Slip');                  //写入页眉文字
$this->Ln(20);                                //换行
}
function Footer(){                            //设置页脚
$this->SetY(-15);                             //设置页脚所在位置
$this->SetFont('Arial','I',8);                //设置页脚字体
$this->Cell(0,10,'Page - '.$this->PageNo());  //输出当前页码作为页脚内容
}
}
$pdf=new PDF('P', 'mm', 'A4');                //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小为 A4
$pdf->Open();                                 //开始创建 PDF
$pdf->AddPage();                              //增加一页
$pdf->SetFont('Arial','',14);                 //设置字体样式
$header=array('Name','Age','Duty','Laborage'); //设置表头
$data=array();                                //设置表体
```



```
$data[0] = array('Simon','24','employee','5,000.00');
$data[1] = array('Elaine','25','Manager','6,000.00');
$data[2] = array('Susan','25','employee','4,000.00');
$data[3] = array('David','26','Manager','8,000.00');
$width=array(40,40,40,40);                          //设置每列宽度
for($i=0;$i<count($header);$i++)                    //循环输出表头
$pdf->Cell($width[$i],6,$header[$i],1);
$pdf->Ln();
foreach($data as $row){                             //循环输出表体
$pdf->Cell($width[0],6,$row[0],1);
$pdf->Cell($width[1],6,$row[1],1);
$pdf->Cell($width[2],6,$row[2],1);
$pdf->Cell($width[3],6,$row[3],1);
$pdf->Ln();
}
$pdf->Output();                                     //输出 PDF 到浏览器
?>
```

运行结果如图 11.15 所示。

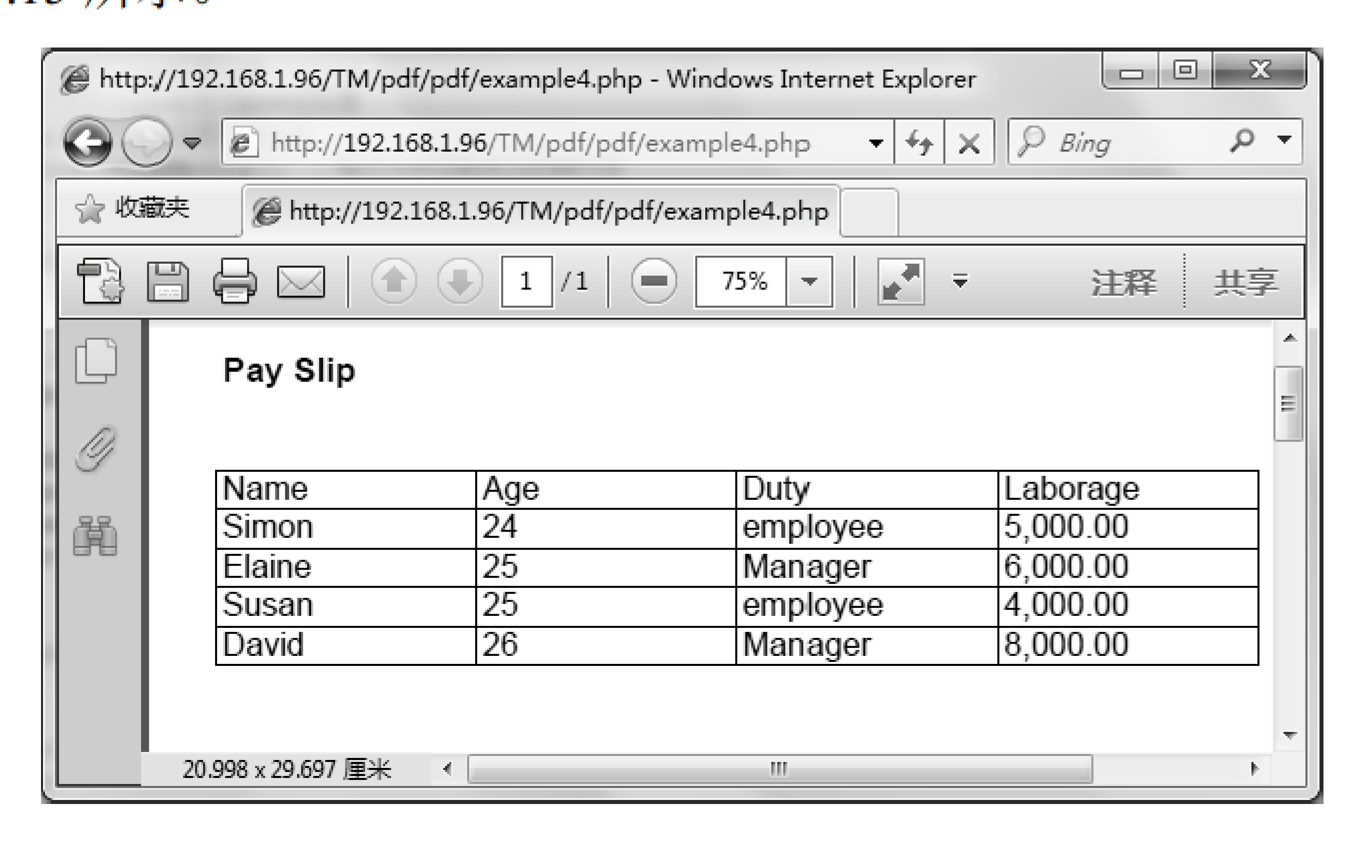


图 11.15　在 PDF 中绘制表格

11.6.6　在 PDF 文档中输出中文

如果应用前面讲解的技术，直接通过 FPDF 在 PDF 文档中输出中文，那么输出的中文字符串将是乱码。解决此问题的方法是应用 FPDF 提供的一个中文插件，继承插件中的 PDF_Chinese 类，在 PDF 文档中输出中文。

首先，下载并解压此插件，复制其中的 chinese.php 文件，将其存储于项目的指定文件夹下；然后通过 include 语句在创建 PDF 文档的文件中载入 chinese.php；最后，创建 PDF 类，继承 PDF_Chinese 类，在 PDF 文档中输出中文。示例 example5.php 的代码如下：

```
<?php
include('pdf/chinese.php');                //载入中文插件
class PDF extends PDF_Chinese{             //继承中文插件类
```

```
function Header(){                              //设置页眉
$this->SetFont('GB','',10);                     //设置字体
$this->Write(10,'FPDF  中文测试');               //设置页眉的内容
$this->Ln(20);                                  //执行换行操作
}
function Footer(){                              //设定页脚
$this->SetY(-15);                               //设置页脚的输出坐标
$this->SetFont('GB','',10);                     //设置字体
$this->Cell(0,10,'第'.$this->PageNo().'页');     //设置页脚输出的内容
}
}
$pdf=new PDF();                                 //创建 PDF 文档
$pdf->AddGBFont();                              //添加字体
$pdf->Open();                                   //开启文档
$pdf->AliasNbPages();                           //为每个页面定义一个别名
$pdf->AddPage();                                //添加页
$pdf->SetFont('GB','I',20);                     //设置字体
$str='山不在高，有仙则名。水不在深，有龙则灵。斯是陋室，惟吾德馨。苔痕上阶绿，草色入廉青。谈笑有鸿儒，
往来无白丁。可以调素琴，阅金经。无丝竹之乱耳，无案牍之劳形。南阳诸葛庐，西蜀子云亭。孔子云：“何陋
之有？”';
$pdf->Write(10,$str);                           //输出中文
$pdf->Output();
?>
```

其运行结果如图 11.16 所示。

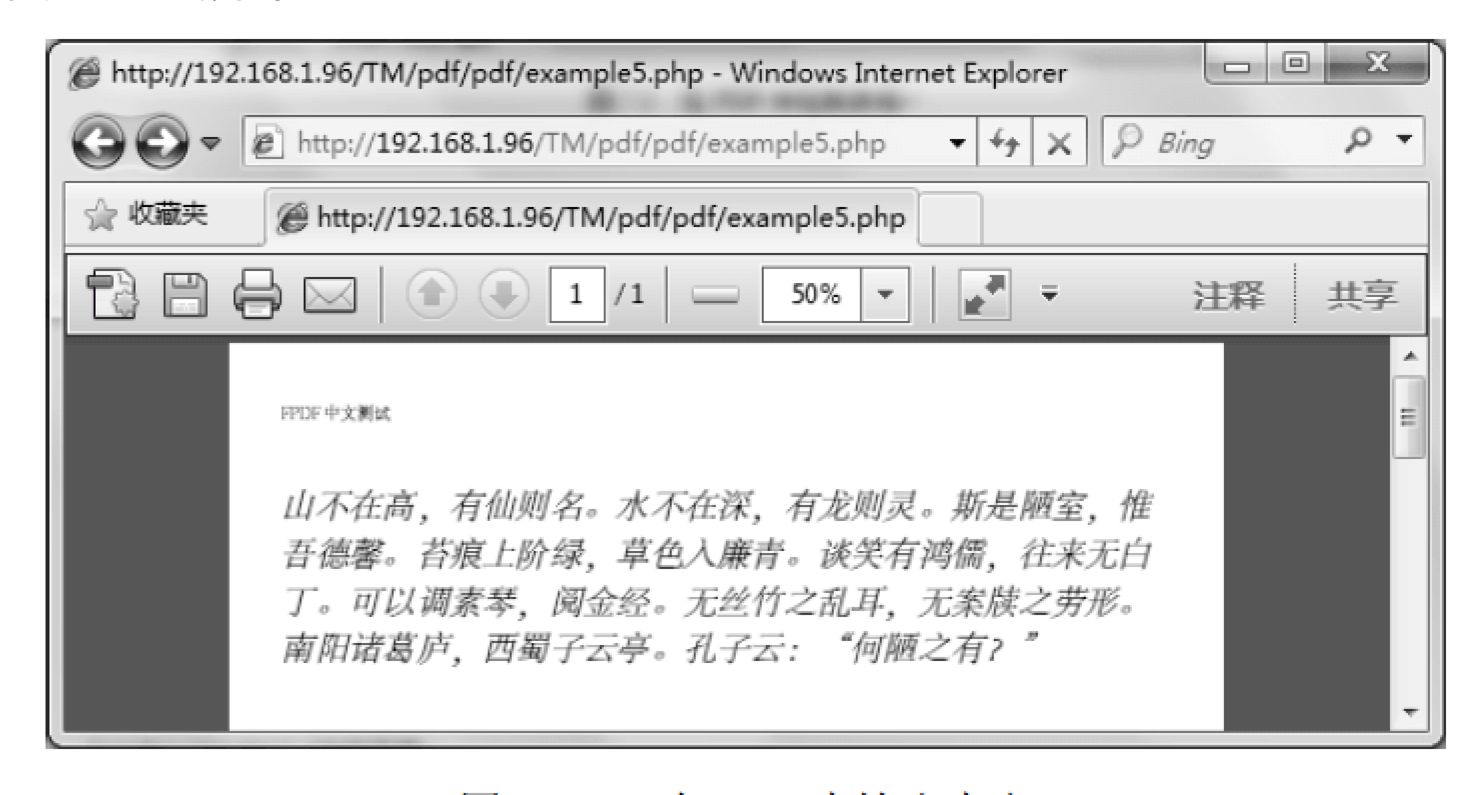


图 11.16　在 PDF 中输出中文

注意 在创建生成 PDF 文档的文件（example5.php）时，页面的编码格式要设置成 gb2312，如果页面的编码格式是 utf-8，那么即使继承了 PDF_Chinese 类，输出的中文仍旧是乱码。

11.6.7　PDF 文档下载

要实现 PDF 文档的下载操作，只需要在通过 Output()函数输出 PDF 文档时，添加两个参数即可。第一个参数定义下载的 PDF 文档的名称；第二个参数设置为 ture，提醒用户下载查看，如果将第二个



参数设置为 false，那么生成的 PDF 文档会直接存储于服务器中，不会提醒用户下载查看。

下面对示例 example5.php 进行修改，将其生成的 PDF 文档命名为“程序测试.pdf”，并且提醒用户下载查看。修改的关键代码如下：

```
<?php
include('pdf/chinese.php');        //载入中文插件
//省略了部分代码
$str='山不在高，有仙则名。水不在深，有龙则灵。斯是陋室，惟吾德馨。苔痕上阶绿，草色入廉青。谈笑有鸿儒，往来无白丁。可以调素琴，阅金经。无丝竹之乱耳，无案牍之劳形。南阳诸葛庐，西蜀子云亭。孔子云：“何陋之有？”  ';
$pdf->Write(10,$str);                //输出中文
$pdf->Output('程序测试.pdf',true);
?>
```

将修改后的示例存储于 example6.php 文件中，运行此示例将输出如图 11.17 所示的 PDF 文件下载页面。



图 11.17 PDF 文档下载

11.7 本章小结

本章运用 FPDF 来创建文档，通过创建 PDF 文档，浏览者可以通过 Adobe Reader 对其内容进行阅读。同时，在创建 PDF 文档的过程中，使整个系统的设计思路更加清晰。通过本章的学习，读者不仅可以了解 PDF 文档的使用，而且可以进行 PDF 文档下载。

第12章

邮件发送系统

（PHP+Zend_Mail 组件实现）

电子邮件是 Internet 上应用最广的服务，通过网络的电子邮件系统，用户可以用极低廉的价格以非常快速的方式与世界上任何一个角落的网络用户联络，这些电子邮件可以是文字、图像、声音等各种形式。电子邮件具有快速、便捷、低成本等优势，适合网络上的信息交流。在构建企业站点时，可以为企业用户提供 E-mail 通信功能，充分利用 E-mail 的功能可以极大地提高企业的工作效率，减轻用户的工作负担。电子邮件的收发技术是 PHP 程序员必须掌握的技术之一。

通过阅读本章，可以学习到：

- **利用 mail()函数发送电子邮件**
- **利用 fsockopen()函数接收电子邮件**
- **带附件邮件的发送与下载**

12.1　邮件发送系统概述

随着 Internet 技术的飞速发展，越来越多的人选择通过网络来进行即时沟通。本章讲解了邮件发送系统从登录邮件服务器，到发送邮件、接收邮件、查看邮件内容以及删除邮件等一系列功能的实现方法。其中还详细地介绍了 WinWebMail 的安装和配置方法。

12.1.1　电子邮件简介

电子邮件（E-mail）是 Internet 上最广泛的应用之一。用户必须拥有因特网服务商（ISP）提供的账户或信箱，才能接收因特网的信件。电子邮件用途十分广泛，可以与世界各地的朋友和客户保持联系，也可以实现收/发、访问等功能。

电子邮件地址的格式是 USER@SERVER.COM，由 3 部分组成：第 1 部分 USER 代表用户信箱的账号，对于同一个邮件接收服务器来说，该账号必须是唯一的；第 2 部分“@”是分隔符；第 3 部分 SERVER.COM 是用户信箱的邮件接收服务器域名，用以标识其所在的位置。

12.1.2　功能结构

邮件发送系统的所有信息都保存在 POP3 服务器上，所以不涉及后台管理及后台数据库。邮件系统的功能结构如图 12.1 所示。

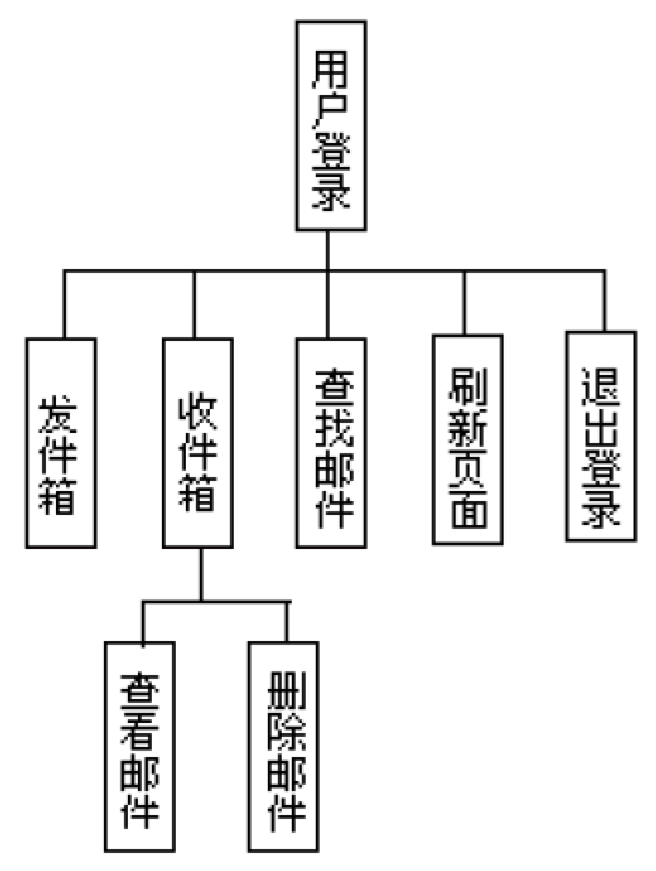


图 12.1　邮件系统的功能结构图

12.1.3　程序预览

为了让读者对本系统有个初步的了解和认识，下面给出本系统的几个页面运行效果图，如果想查看完整的效果图，请参见光盘源程序。

登录邮件服务器的页面如图 12.2 所示。



图 12.2　登录邮件服务器的页面

在邮件发送系统中发送带附件邮件的页面运行效果如图 12.3 所示。



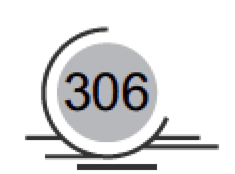
图 12.3　发送带附件邮件的页面运行效果

邮件发送系统中收件箱的运行效果如图 12.4 所示。



图 12.4　收件箱的运行效果

邮件发送系统中查看邮件详细信息的页面运行效果如图 12.5 所示。



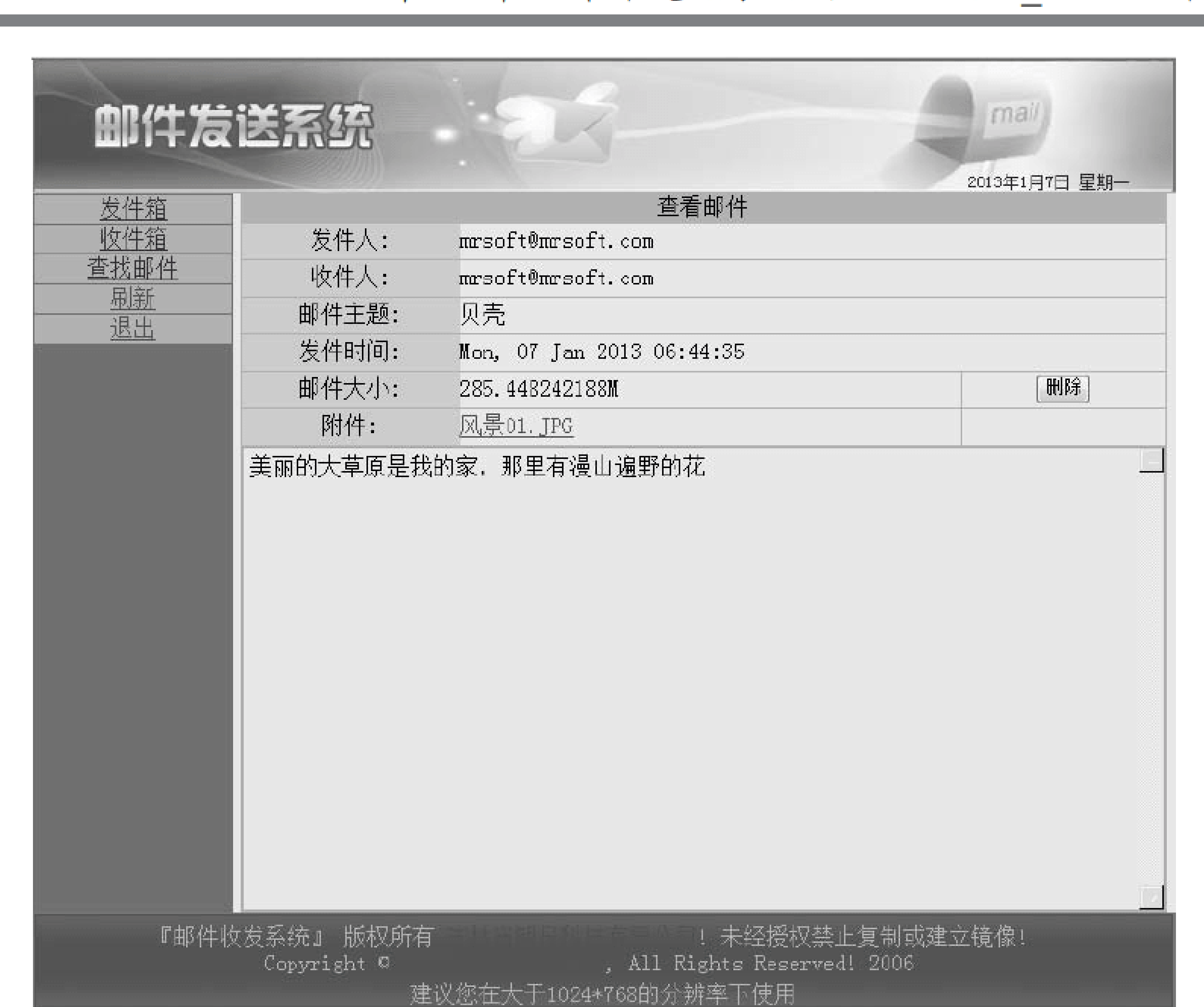


图 12.5　查看邮件详细信息的页面运行效果

12.2　邮件收发的原理

在具体讲解邮件发送系统功能之前，首先介绍一下邮件发送系统发送邮件、接收邮件、查找邮件及删除邮件实现的基本原理。

☑ 发送邮件实现相对简单，用 PHP 自身提供的 mail()函数就可以完成该功能。需要注意的是，由于 mail()函数自身不能进行 SMTP 认证，因此需要安装一个 SMTP 服务器转发邮件，并且需要在 php.ini 中设置 SMTP 服务器地址及端口号。

☑ 接收邮件需要利用 fsockopen()函数来实现，首先利用该函数连接 POP3 服务器并返回一个 socket 句柄，之后就可以利用 fputs()函数将 POP 命令发送给服务器，利用 fgets()函数接收 POP 服务器反馈回来的信息（如邮件头、邮件内容），从邮件头中就可以提取发件人地址、发件时间等信息。

☑ 查找邮件时，系统首先遍历所有 POP 服务器中的信件，并提取所有信件的主题和发件人地址，如果发现与用户查找的主题或发件人地址相同的邮件，就将该邮件显示出来。

☑ 删除邮件只需用 fputs()函数将 POP 命令 DELE 写入服务器，但需注意删除邮件必须由命令 QUIT 来执行，否则即使发送了 DELE 命令也无法完成删除工作。

以上内容为整个邮件系统实现的核心思想，笔者将接收邮件的各项功能封装到类 POP3 中，这样做的目的既符合 PHP 面向对象的发展趋势，又为系统带来了很好的移植性，并且为日后的维护工作带来很大方便。

12.3 通过 WinWebMail 配置邮件服务器

在本模块中，通过第三方的 WinWebMail 邮件服务器，在本地测试电子邮件系统的功能。下面介绍一下 WinWebMail 的安装和配置。

（1）打开如图 12.6 所示的 WinWebMail 3.7.7.1 标准版安装界面，单击 Next 按钮。

（2）进入如图 12.7 所示的 WinWebMail 3.7.7.1 标准版协议界面，单击 Next 按钮。

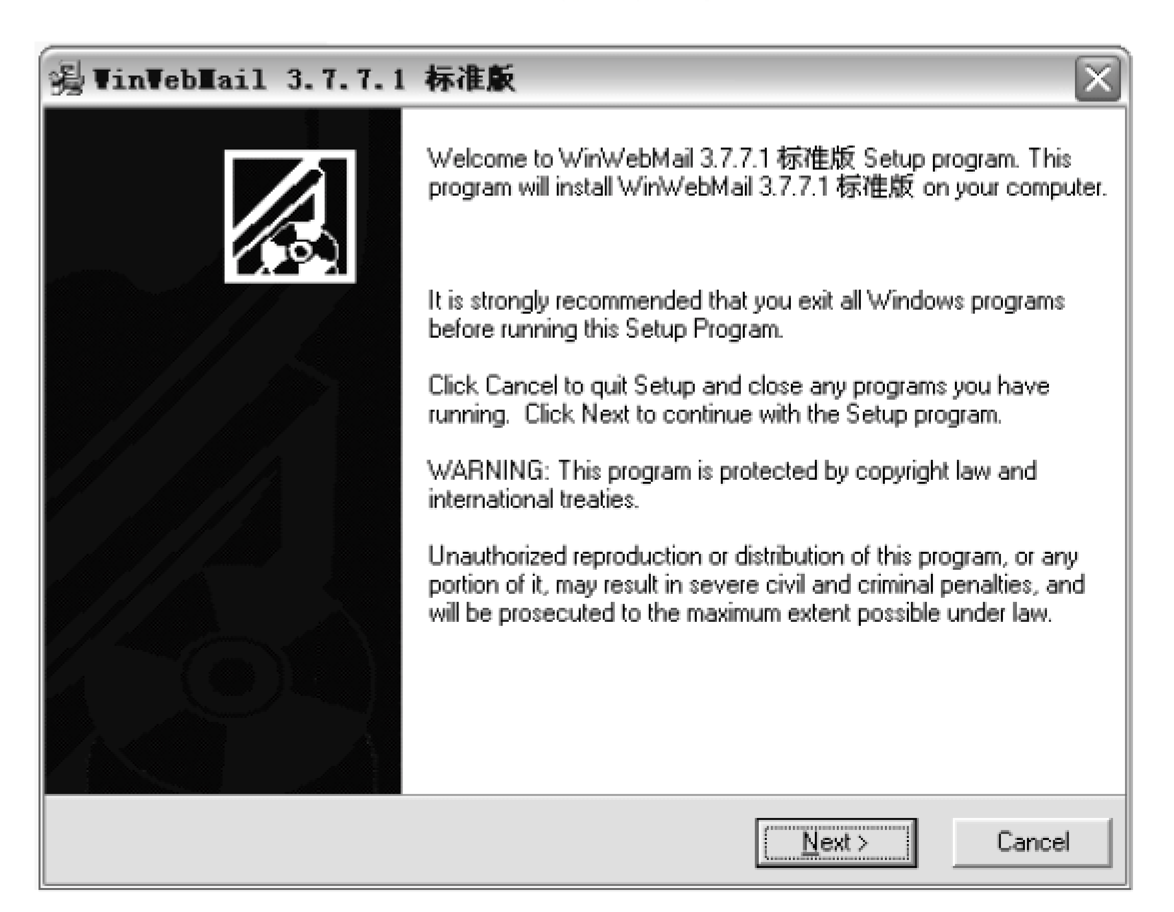


图 12.6　WinWebMail 3.7.7.1 标准版安装界面

图 12.7　WinWebMail 3.7.7.1 标准版协议界面

（3）进入如图 12.8 所示的 WinWebMail 3.7.7.1 标准版安装路径界面，可以选择所要安装的路径，单击 Next 按钮。

（4）进入如图 12.9 所示的 WinWebMail 3.7.7.1 标准版软件名称界面，单击 Next 按钮进入如图 12.10 所示的安装完毕界面，单击 Finish 按钮安装完毕。

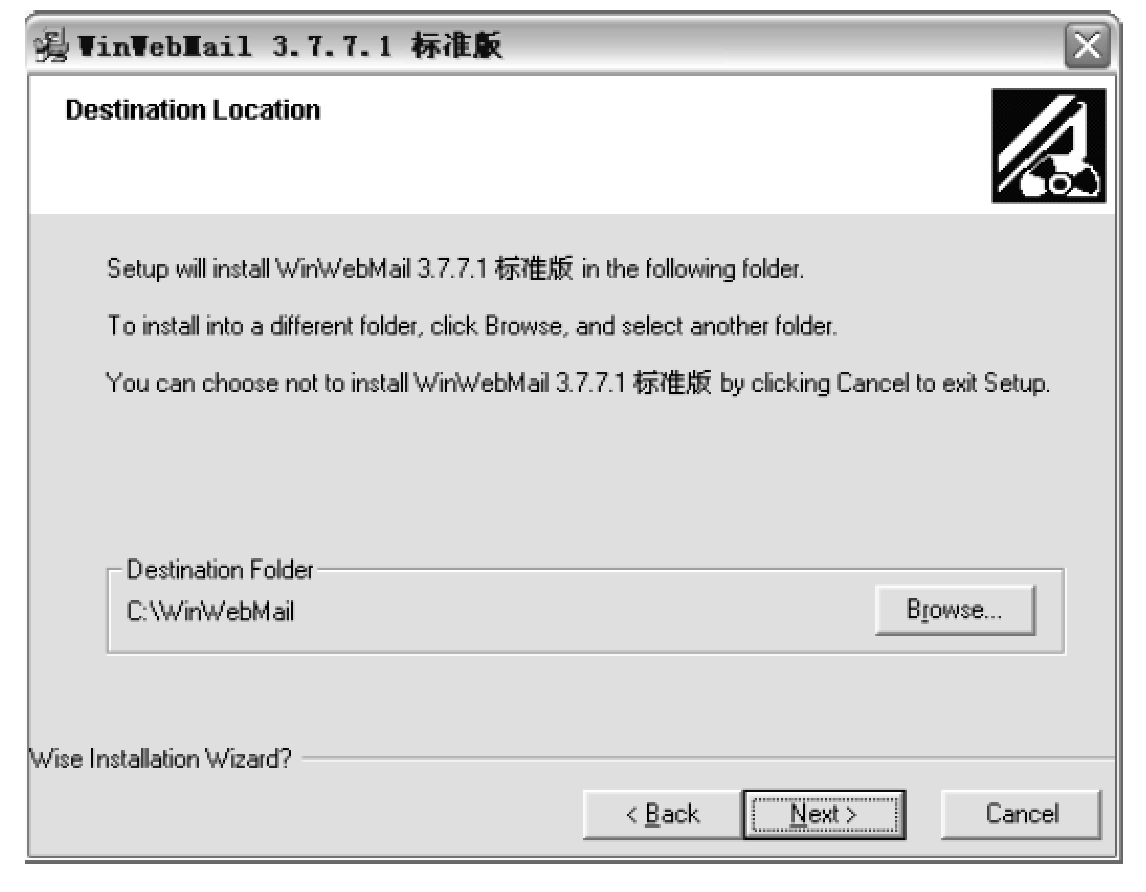


图 12.8　WinWebMail 3.7.7.1 标准版安装路径界面

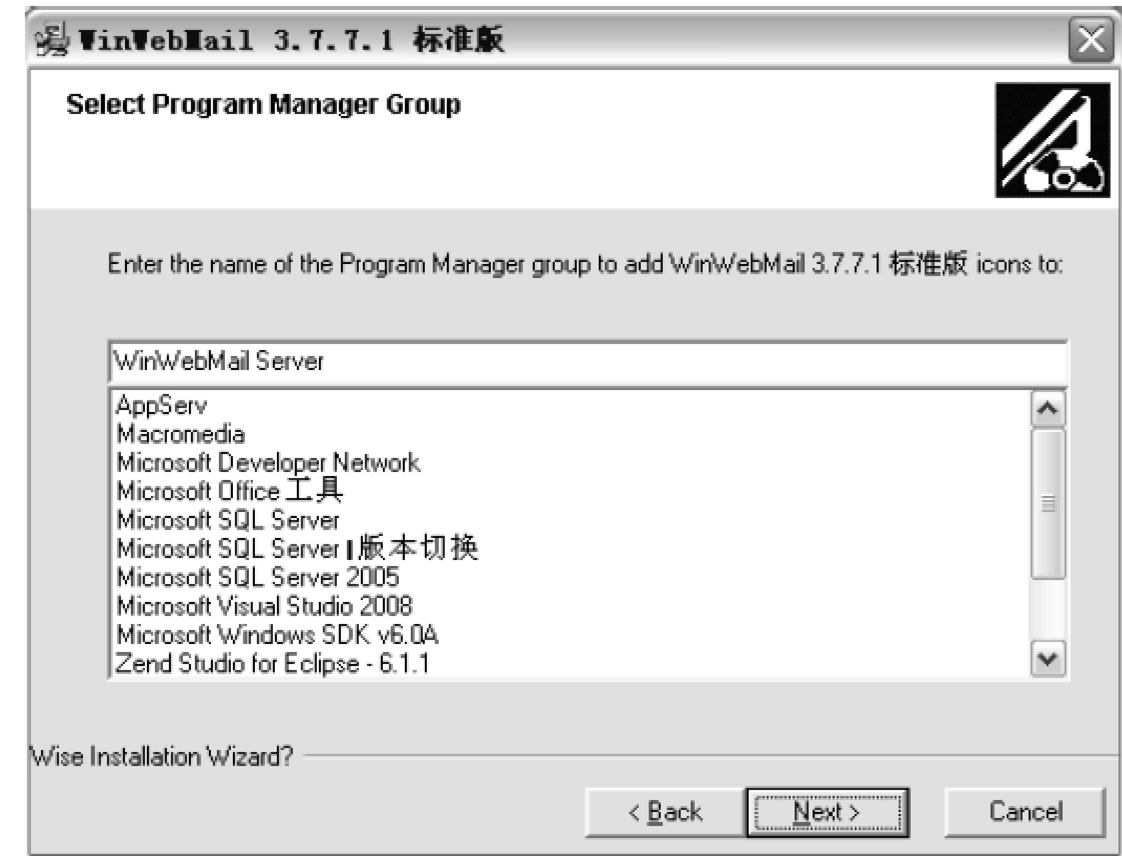
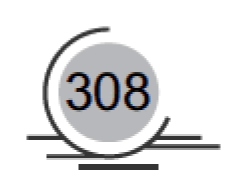


图 12.9　WinWebMail 3.7.7.1 标准版软件名称界面



（5）启动 WinWebMail，用户可以单击右下角带红色 X 的蓝色图标，在弹出的快捷菜单中选择“服务”命令，如图 12.11 所示。

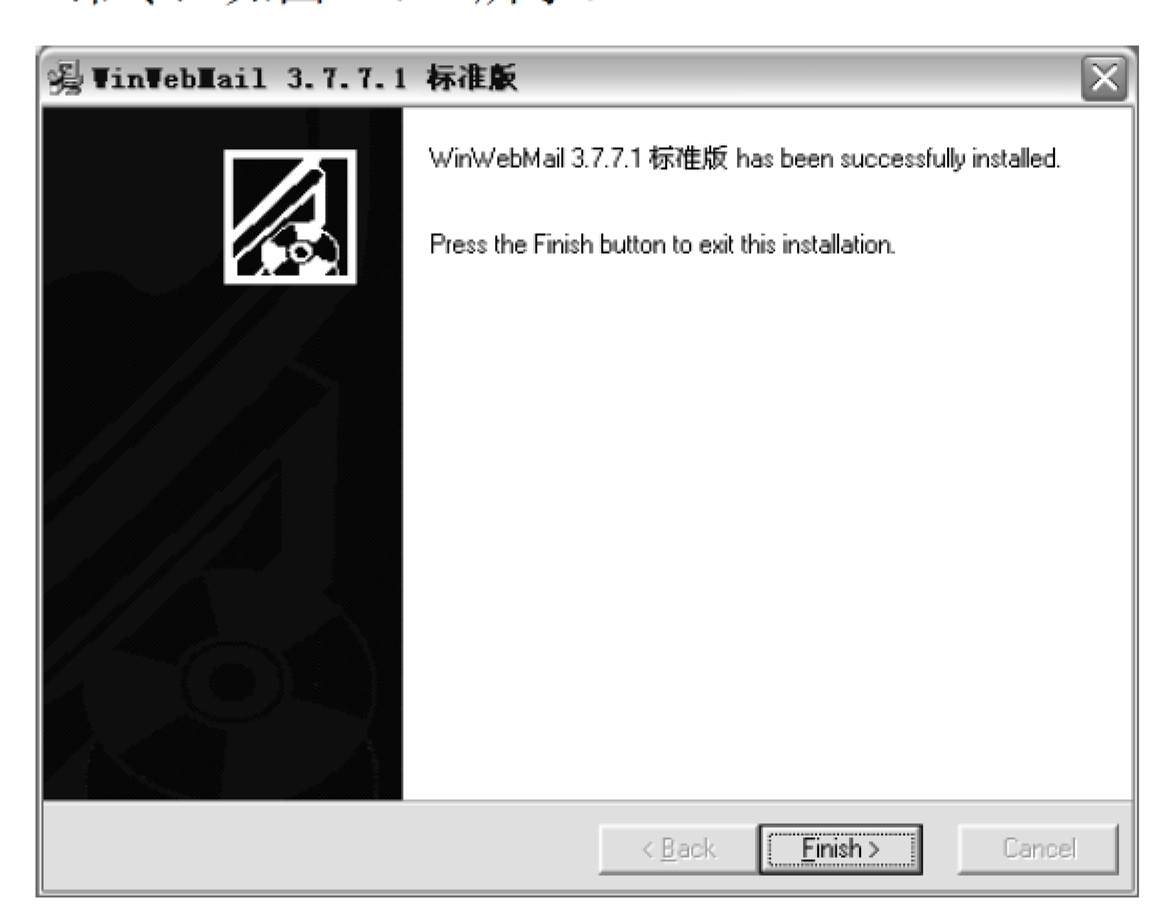


图 12.10　WinWebMail 3.7.7.1 标准版安装完成界面

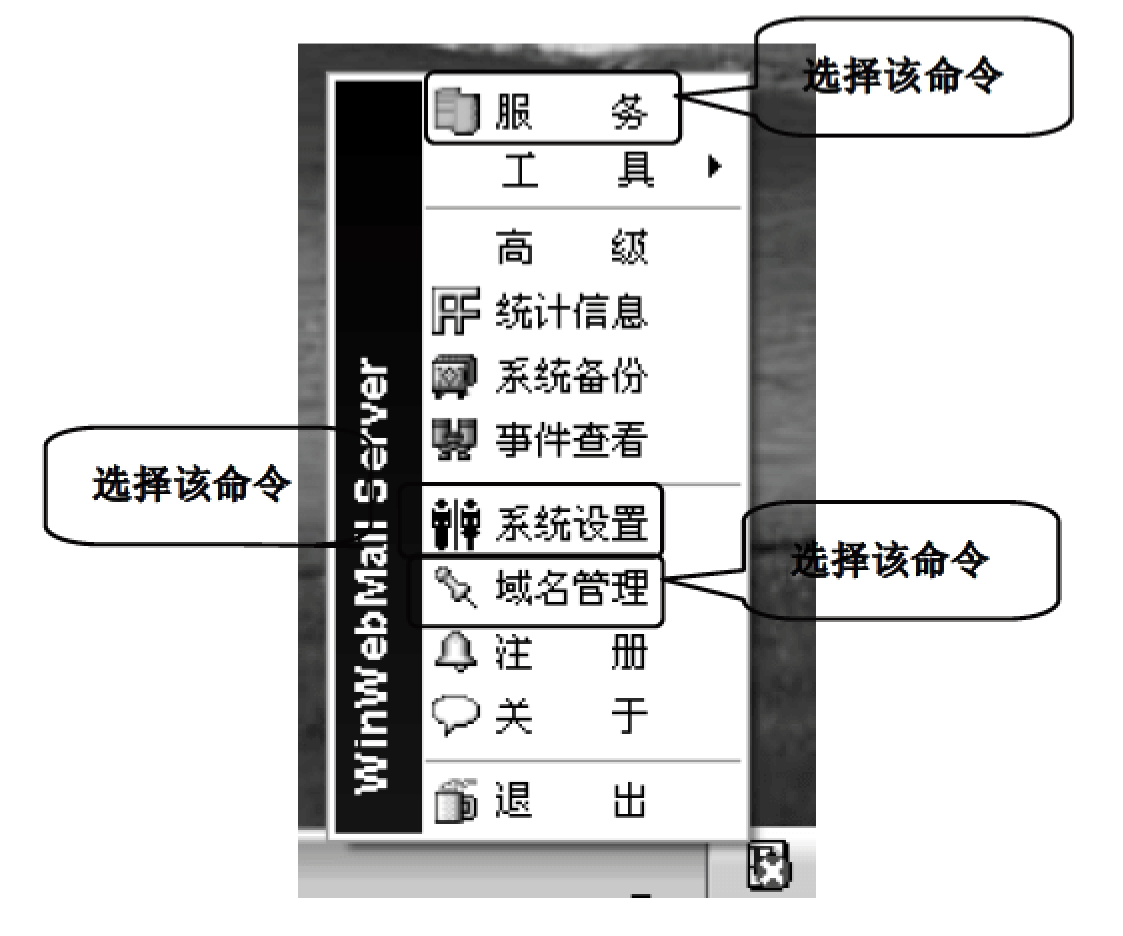


图 12.11　WinWebMail 选项

（6）当选择“服务”命令后进入如图 12.12 所示的服务界面，选中“修改”复选框，输入 DNS 的 IP 地址，单击“启动 WinWebMail 服务程序”按钮，再单击绿色对号图标，服务选项配置完成。

（7）重新进入图 12.11 所示的操作界面，选择“域名管理”命令，弹出如图 12.13 所示的界面。添加域名后，单击绿色对号图标。

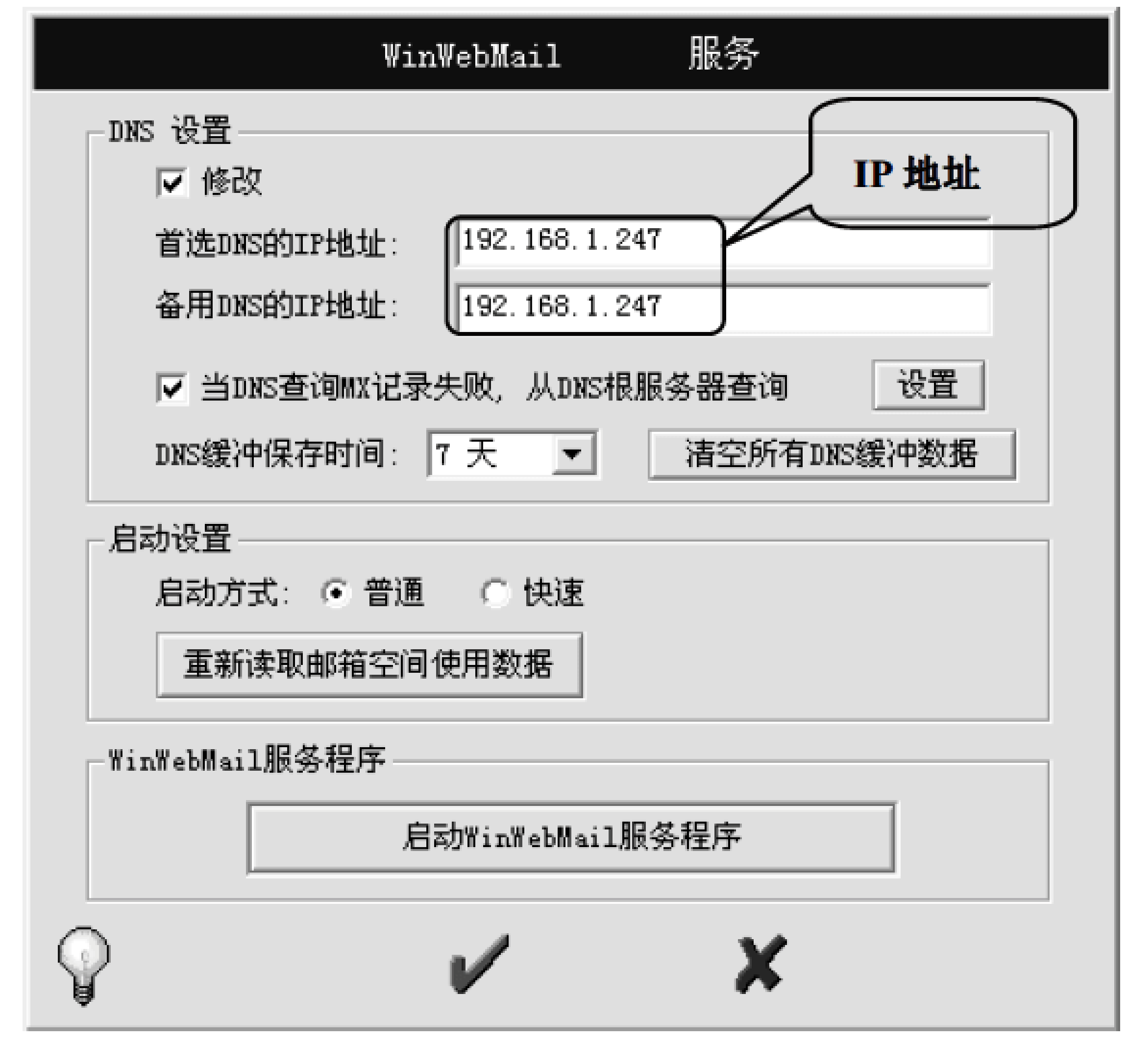


图 12.12　WinWebMail 服务界面

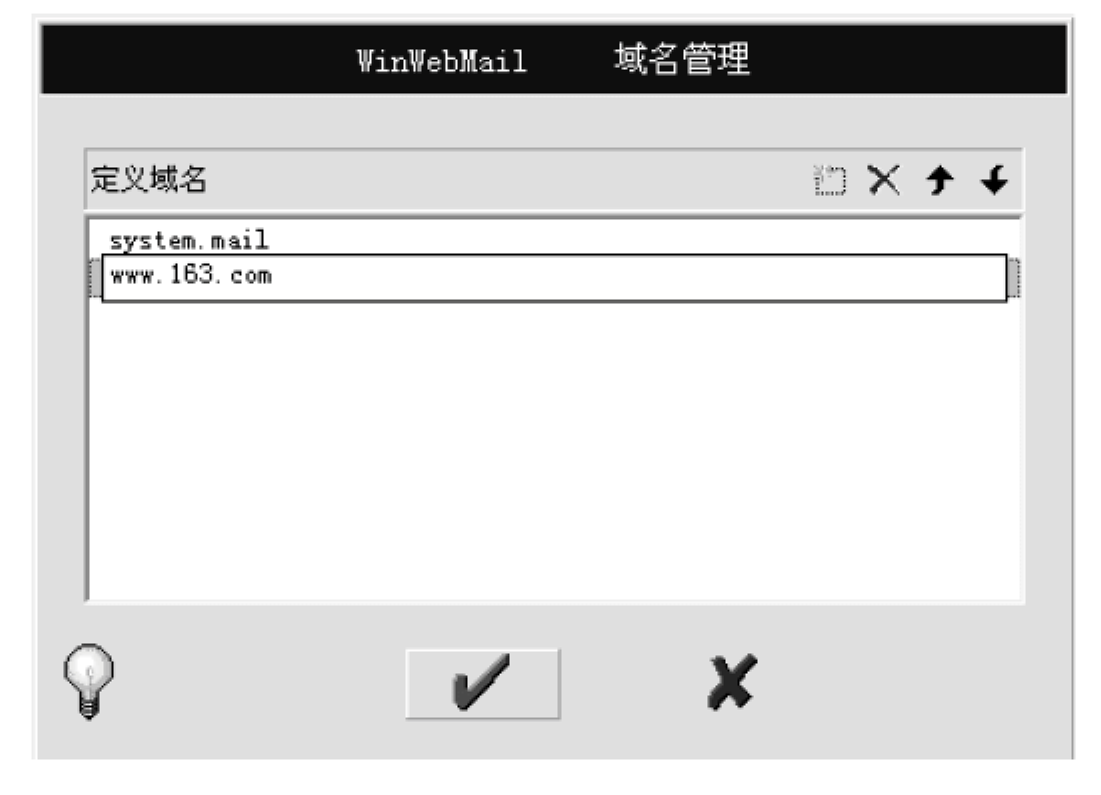


图 12.13　WinWebMail 域名管理

（8）重新进入图 12.11 所示的界面，选择“系统设置”命令，弹出如图 12.14 所示的界面。添加用户名、密码，选择所在域名后单击绿色对号图标。

至此，WinWebMail 配置完毕。



图 12.14　WinWebMail 系统设置

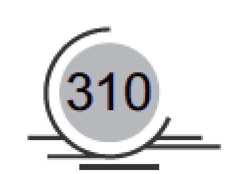
12.4　登录邮件服务器

用户登录邮件服务器时，在如图 12.15 所示的登录界面中输入 POP3 服务器地址、用户在 POP3 服务器注册的邮箱名称和密码，单击“登录”按钮，将数据提交到 chklogin.php 文件中，完成与服务器的连接操作。



图 12.15　邮件服务器登录

在 chklogin.php 文件中，获取表单提交的服务器地址、邮箱名称和密码，调用 POP3 类中的 open() 和 login()方法，完成与服务器的连接，并且将登录成功后的服务器地址、邮箱名称和密码赋给指定的





Session 变量。关键代码如下：

```
<?php
session_start();
include("mailclass.php");                    //包含发邮件类
$hostname=$_POST['servername'];              //获取提交的服务器地址
$mailname=substr($_POST['mailname'],0,strpos($_POST['mailname'],'@'));        //获取提交的用户邮箱地址
$mailpwd=$_POST['mailpwd'];                  //获取提交的用户邮箱密码
$rec=new pop3($hostname,110,10);             //对发邮件类进行初始化
if(!$rec->open()){     //如果调用该类的 open()方法返回 false 值，则说明没有成功连接上服务器，这时给出提示
echo"<script>alert('对不起，无法连接服务器！');history.back();</script>";
exit;
}
//如果调用发邮件类的 login()方法返回 false 值，则说明用户输入的名称或密码错误
if(!$rec->login($mailname,$mailpwd)){
echo"<script>alert('对不起，您的用户名或密码输入错误！');history.back();</script>";
exit;
}
$_SESSION['host']=$hostname;//如果用户登录成功，则注册 Session 变量 host 保存服务器地址，以备其他页面使用
//如果用户登录成功，则注册 Session 变量 user 保存用户邮箱地址，以备其他页面使用
$_SESSION['user']=$_POST['mailname'];
$_SESSION['pass']=$mailpwd;//如果登录成功，则注册 Session 变量 pass 保存用户邮箱密码，以备其他页面使用
$rec->close();                               //调用发邮件类的 close()方法关闭与 POP3 服务器的连接
header("location:list.php");                 //定位到发件箱页面
?>
```

12.5　邮 件 收 发

12.5.1　邮件收发模块概述

成功登录邮件服务器之后，就可以完成邮件的发送、接收、查看、查找和删除操作。邮件收发模块的功能如图 12.16 所示。

在发件箱中可以完成无附件邮件和有附件邮件的发送操作；在收件箱中可以查看接收到的邮件主题，以及邮件的具体内容，同时还可以完成附件的下载操作；在查找邮件中可以根据邮件主题和发件人对邮件进行查找；刷新功能完成对页面的刷新操作，以此更新出邮箱最新收到的邮件信息；退出实现当前登录用户的退出操作。



图 12.16　邮件收发模块的功能

12.5.2　发送邮件

发送邮件的操作由 list.php 和 sendmail.php 两个文件完成。在 list.php 中定义发送邮件表单，如图 12.17 所示。

图 12.17　发送邮件

为了缓解服务器压力，在添加发送邮件的信息时，应用 JavaScript 脚本对表单提交的数据进行验证，其中发件人、收件人和邮件内容为必填内容，并且格式必须正确，而附件为可选项，抄送和密送不必填写。验证表单提交数据格式是否正确使用的是 JavaScript 脚本的自定义函数 chkinput()，该函数的部分代码如下：

```
<!—*********************************list.php************************************-->
<script language="JavaScript">
function chkinput(form)                    //定义 chkinput()函数，用来验证用户输入的信息
{
if(form.mailfrom.value=="")                //判断用户输入的信息是否为空
{
//如果用户输入的信息为空，则利用 window 的 alert()方法弹出一个提示框，告诉用户某些必填的内容没有填写
alert("请输入发件人地址!");
form.mailfrom.select();                    //选择当前窗体元素
return(false);                             //返回 false 值使表单不能提交
}
…                                          //省略了功能类似的代码，请见本书附带光盘
//由于电子邮箱地址中一定含有字符“@”，所以通过用户输入的邮箱地址是否含有字符“@”就可以判断用户输入的邮箱地址是否正确，函数 indexOf()的作用是判断该函数参数在某字符串中第一次出现的位置，如果结果小于0，则说明没有查找到该字符
if(form.mailfrom.value.indexOf('@')<0)
{
alert("发件人 E-mail 格式输入错误!");
form.mailfrom.select();
return(false);
}
…
return(true);                              //如果用户输入的信息通过验证，则该函数返回 true，从而使表单能够提交
}
</script>
```

在 sendmail.php 文件中，首先获取表单提交的邮件内容。然后对邮件标题和内容进行编码格式转换。最后根据附件的内容进行判断，如果附件为空，则直接应用 mail()函数完成邮件的发送；如果附件





不为空，则需要通过文件系统函数读取附件的内容，对其编码格式进行转换，定义到邮件体中后，应用 mail()函数完成邮件的发送。关键代码如下：

```
<!—********************************sendmail.php***********************************-->
<?php
include("function.php");
$mailfrom=$_POST['mailfrom'];                          //获取发件人地址
$mailto=$_POST['mailto'];                              //获取收件人地址
$mailcc=$_POST['mailcc'];                              //获取抄送人地址
$mailbcc=$_POST['mailbcc'];                            //获取密送人地址
$mailsubject=base64_encode($_POST['mailsubject']);     //获取邮件主题
$mailbody = base64_encode($_POST['mailbody']);         //获取邮件内容
if(empty($_FILES['upload_file']['name'])){
$headers = "From:$mailfrom \r\n Cc:$mailcc \r\n Bcc:$mailbcc"; //定义邮件头
$headers .= "Reply-To:$mailfrom\r\n";
$headers .= "Content-type: text/html;\r\n charset=iso-8859-1 \r\n";
if(@mail($mailto, $mailsubject,$mailbody,$headers)){   //通过 mail()函数发送邮件，并给出邮件发送结果
echo "<script>alert('邮件发送成功！');history.back()</script>";
}else{
echo "<script>alert('邮件发送失败！');history.back()</script>";
}
}else{
$file = $_FILES['upload_file']['tmp_name'];            //获取附件名称
$boundary = uniqid("");                                //定义分界线
$headers = "MIME-Version: 1.0 \n";
$headers .= "Content-type: multipart/mixed; Boundary= $boundary \r\n";
$headers .= "From:$mailfrom\r\n";                      //定义邮件头
if($_FILES['upload_file']['type']){                    //判断附件类型
$mimeType = $_FILES['upload_file']['type'];
}else{
$mimeType ="application/unknown";                      //获取附件类型
}
$fileName = $_FILES['upload_file']['name'];            //获得附件名称
$fp = fopen($file, "r");                               //打开文件
$read = fread($fp, filesize($file));                   //将附件读入变量$read
$read = base64_encode($read);                          //转换为 base64 编码
$read = chunk_split($read);                            //将长字符串切成由每行 76 个字符组成的小块
/*定义邮件体*/
$body = "--$boundary--
Content-type: text/plain; charset=iso-8859-1
$mailbody
--$boundary--
Content-type: $mimeType; name=$fileName
Content-disposition: attachment; filename=$fileName
Content-transfer-encoding: base64
$read
--$boundary--";
if(mail($mailto, $mailsubject,$body,$headers)){        //用 mail()函数发送邮件
echo "<script>alert('附件发送成功！');history.back()</script>";
```

```
}else{
echo "<script>alert('附件发送失败！');history.back()</script>";
}
}
?>
```

说明 当所发送的邮件中带有附件时，首先需要将邮件的 MIME 的 Content-type 类型定义成 multipart/mixed，然后定义邮件内容分割线（一串字符或数字），最后将文件内容编写在邮件主体内容中进行发送。

注意 PHP 的 mail()函数只能将邮件发送到 SMTP 的邮件转发器中，然后通过转发器将邮件发送出去。

12.5.3 接收邮件

接收邮件的核心是调用 POP3 类中的 open()、login()、stat()和 listmail()方法，实现邮件的接收操作。POP3 类独立存储于 mailclass.php 中，将在 12.6 节中进行详细讲解。这里将具体介绍系统是如何调用 POP3 类实现收邮件功能的，收件箱页面如图 12.18 所示。

邮件发送系统

2013年1月7日 星期一

发件箱 收件箱 查找邮件 刷新 退出

收件箱

	主题	发件人	时间	大小
×	逆	333	Wed, 26 Dec 2012 16:04:21	482KB
×	氘	666	Wed, 26 Dec 2012 16:04:32	482KB
×	小可爱	mrsoft@mrsoft.com	Sat, 05 Jan 2013 05:18:43	4.55M
×	词典	mrsoft@mrsoft.com	Sat, 05 Jan 2013 05:21:50	1175M
×	图书	mingri@mingri.com	Sat, 05 Jan 2013 05:23:24	544KB
×	贝壳	mrsoft@mrsoft.com	Mon, 07 Jan 2013 06:44:35	285.M

您的邮箱中共有邮件：6 封占用空间：1467M

『邮件收发系统』 版权所有 ！未经授权禁止复制或建立镜像！
Copyright © , All Rights Reserved! 2006
建议您在大于1024*768的分辨率下使用

图 12.18　收件箱页面

在 receivemail.php 文件中，首先初始化 Session 变量，判断当前用户是否具有访问权限；然后通过 include 语句载入 POP3 类文件 mailclass.php；最后实例化 POP3 类，通过返回的对象调用类中的方法，循环输出收件箱中的邮件信息。关键代码如下：





```
<!--*********************************receivemail.php**********************************-->
<?php
session_start();
if(!isset($_SESSION['user'])){
echo "<script>alert('请不要非法登录本站！');history.back();</script>";
exit;
}                              //以上代码通过判断 Session 变量 user 的值是否为空来判断用户是否非法登录
include("mailclass.php");                    //包含发邮件类
$rec=new pop3($_SESSION['host'],110,10);     //对邮件类实例化
$rec->open();                                //调用邮件类的 open()方法来与 POP3 服务器建立连接
//调用邮件类的 login()方法来验证用户身份
$rec-> login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']);
$rec->stat();                                //调用邮件类的 stat()方法来获得邮件总数和邮件总字节数
$rec->listmail();                            //调用邮件类的 listmail()方法获得每个邮件的大小及序号
?>……
<?php
if($rec->messages<=0){                       //判断邮件记录数
?>
<table width="650" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td bgcolor="#66CCFF"><div align="center">您的收件箱内无邮件！</div></td>
</tr>
</table>
<?php
}else{                                       //如果邮件存在，则执行下面的操作
?>
<table width="650" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="40"   bgcolor="#66CCFF">
<table width="650" height="40" border="0" cellpadding="0" cellspacing="1">
<tr>
<td height="20" colspan="2" bgcolor="E9E8E8"><div align="center">主题</div></td>
<td width="136" bgcolor="E9E8E8"><div align="center">发件人</div></td>
<td width="183" bgcolor="E9E8E8"><div align="center">时间</div></td>
<td width="95" bgcolor="E9E8E8"><div align="center">大小</div></td>
</tr>
<?php
for($i=1;$i<=$rec->messages;$i++){           //循环输出邮件数据
$rec->getmail($i);
?>
<tr>
<td width="43" height="20" bgcolor="E9E8E8"><div align="center">
<a href="deletemail.php?del=<?php echo $i;?>"><img src="images/drop.png" border="0"></a>
</div></td>
<td width="187" bgcolor="E9E8E8"><div align="left">
<a href="readmail.php?mailid=<?php echo $i;?>">
<?php
for($j=0;$j<count($rec->head);$j++){         //通过循环提取所有保存在 head 数组中的邮件头的信息
if(substr($rec->head[$j],0,20)=="Subject: =?GB2312?Q?"){     //获取邮件主题
```

```
echo quoted_printable_decode(substr(htmlspecialchars($rec->head[$j]),20,
strlen(trim(htmlspecialchars($rec->head[$j])))-22))."<br>\n";            //输出邮件主题
}else if(substr($rec->head[$j],0,7)=="Subject"){                          //去掉邮件主题的提示文字 Subject
echo base64_decode(substr(htmlspecialchars($rec->head[$j]),8,
strlen(trim(htmlspecialchars($rec->head[$j])))-8))."<br>\n";
}
}
?>
</a>
</div></td>
<td height="20" bgcolor="E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){
if(substr($rec->head[$j],0,4)=="From"){                                  //获取发件人地址
//去掉发件人地址的提示文字 From
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
}
}
?>
</div></td>
<td height="20" bgcolor="E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){                                     //获取头文件数据
if(substr($rec->head[$j],0,4)=="Date"){                                  //截取时间数据
echo  substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";// 输
出时间
}
}
?>
</div></td>
<td height="20" bgcolor="E9E8E8"><div align="center">
<?php
$size=$rec->mail_list[$i]['size'];                                       //获取文件大小
if($size>=1024){                                                         //计算文件大小，以不同单位进行换算输出
echo substr(($size/1024),0,4)."M";
}elseif($size>1024*1024){
echo substr(($size/(1024*1024)),0,4)."G";
}elseif($size<1024){
echo ($size)."KB";
}
?>
</div></td>
</tr>
<?php
$rec->head=NULL;
}
?>
</table>
</td>
```

```
</tr>
</table>
<table width="650" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="114"></td>
<td width="536"><div align="right">您的邮箱中共有邮件: <?php echo $rec->messages;?> 封占用
空间:
<?php
$totalsize=$rec->size;                                   //计算邮件服务器空间大小
if($totalsize>=1024){
echo substr(($totalsize/1024),0,4)."M";
}elseif($totalsize>1024*1024){
echo substr(($totalsize/(1024*1024)),0,4)."G";
}elseif($totalsize<1024){
echo ($totalsize)."KB";
}
?>
 </div></td>
</tr>
</table>
<?php
$rec->close();                                           //关闭服务器
}
?>
```

12.5.4　查看邮件详细内容

在收件箱的 receivemail.php 文件中，单击邮件主题超链接，将跳转到 readmail.php 页中，在该文件中根据超链接传递的邮件 id 值，输出指定邮件的详细信息。其仍然是调用 POP3 类中的方法，完成邮件详细信息的输出操作。关键代码如下：

```
<?php
session_start();                                         //初始化 Session 变量
if(!isset($_SESSION['user'])){                           //判断用户的权限
echo "<script>alert('请不要非法登录本站！');history.back();</script>";
exit;
}
$mailid=$_GET['mailid'];                                 //获取超链接传递的邮件 ID 值
include("mailclass.php");                                //载入 POP3 类
$rec=new pop3($_SESSION['host'],110,10);                 //实例化 POP3 类
$rec->open();                                            //连接邮件服务器
$rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']);  //用户登录
$rec->stat();                                            //验证用户是否正确登录
$rec->listmail();                                        //读取邮件
$rec->getmail($mailid);                                  //根据超链接传递的 id 值，获取指定邮件的内容
?>
<table width="650" border="0" align="center" cellpadding="0" cellspacing="1">
<tr>
```

```
<td width="152" height="25" bgcolor="#CCCCCC"><div align="center">发件人:</div></td>
<td colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){                                    //获取发件人信息
if(substr($rec->head[$j],0,4)=="From"){                                 //输出发件人
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
}
}
?>
</div></td>
</tr>
<tr>
<td height="25" bgcolor="#CCCCCC"><div align="center">收件人:</div></td>
<td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left"><?php echo $_SESSION['user']; ?></div></td>
</tr>
<tr>
<td height="25" bgcolor="#CCCCCC"><div align="center">邮件主题:</div></td>
<td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){                                    //获取邮件主题
if(substr($rec->head[$j],0,20)=="Subject: =?GB2312?Q?"){                //对邮件主题进行截取
echo quoted_printable_decode(substr(htmlspecialchars($rec->head[$j]),20,
strlen(trim(htmlspecialchars($rec->head[$j])))-22))."<br>\n";           //输出邮件主题
}else if(substr($rec->head[$j],0,7)=="Subject"){
echo base64_decode(substr(htmlspecialchars($rec->head[$j]),8,
43    strlen(trim(htmlspecialchars($rec->head[$j])))-8))."<br>\n";
}
}
?>
</div></td>
</tr>
<tr>
<td height="25" bgcolor="#CCCCCC"><div align="center">发件时间:</div></td>
  <td height="20" colspan="2" bgcolor="#E9E8E8"><div align="left">
<?php
for($j=0;$j<count($rec->head);$j++){                                    //获取发件时间
if(substr($rec->head[$j],0,4)=="Date"){                                 //对发件时间进行截取输出
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";
}
}
?>
</div></td>
</tr>
<tr>
<td height="25" bgcolor="#cccccc"><div align="center">邮件大小:</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="left">
<?php
$size=$rec->mail_list[$mailid]['size'];                                 //获取邮件大小
if($size>=1024){                                                        //对邮件大小进行换算输出
```

```
echo ($size/1024)."M";
}elseif($size>1024*1024){
echo ($size/(1024*1024))."G";
}elseif($size<1024){
echo ($size)."KB";
}
if(isset($rec->body[6])){                                    //判断是否存在附件
$file='';
$filename=substr(strrchr($rec->body[6],"="),1,200);          //获取附件名称
for($i=0;$i<count($rec->body);$i++){                         //读取附件数据
if($i>7 && $i<count($rec->body)-2){
$file=base64_decode($rec->body[$i]);                         //对附件进行编码转换
}
}
file_put_contents("file/".$filename,$file);                  //将附件数据写入到指定文件中
}
?>
</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<input type="button" value="删除" class="buttoncss"
onClick="javascript:window.location='deletemail.php?del=<?php echo $mailid;?>&filename=<?php echo $filename;?>'">
</div></td>
</tr>
<?php
if(isset($rec->body[6])){                                    //判断附件是否存在，存在则创建附件下载超链接
?>
<tr>
<td height="25" bgcolor="#cccccc"><div align="center">附件:</div></td>
<td width="352" height="20" bgcolor="#E9E8E8"><?php echo '
<a href="down.php?filename='.$filename.'" target="_blank">'.$filename.'</a>'; ?>
</td>
<td width="142" bgcolor="#E9E8E8"> </td>
</tr>
<?php
}
?>
<tr>
<td height="325" colspan="3" bgcolor="#E9E8E8" valign="top">
        <?php
if(isset($rec->body[6])){
echo base64_decode($rec->body[3])."<br>";
}else{
if(substr($rec->body[1],0,1)!="="){
echo base64_decode($rec->body[1]);
}else{
echo quoted_printable_decode($rec->body[1]);
}
}
?>
```

```
</td>
</tr>
</table>
```

12.5.5 下载附件

下载附件的操作从查看邮件内容中的附件超链接开始，链接到 down.php 文件，在该文件中根据超链接传递的附件名称实现附件下载的操作。其运行结果如图 12.19 所示。

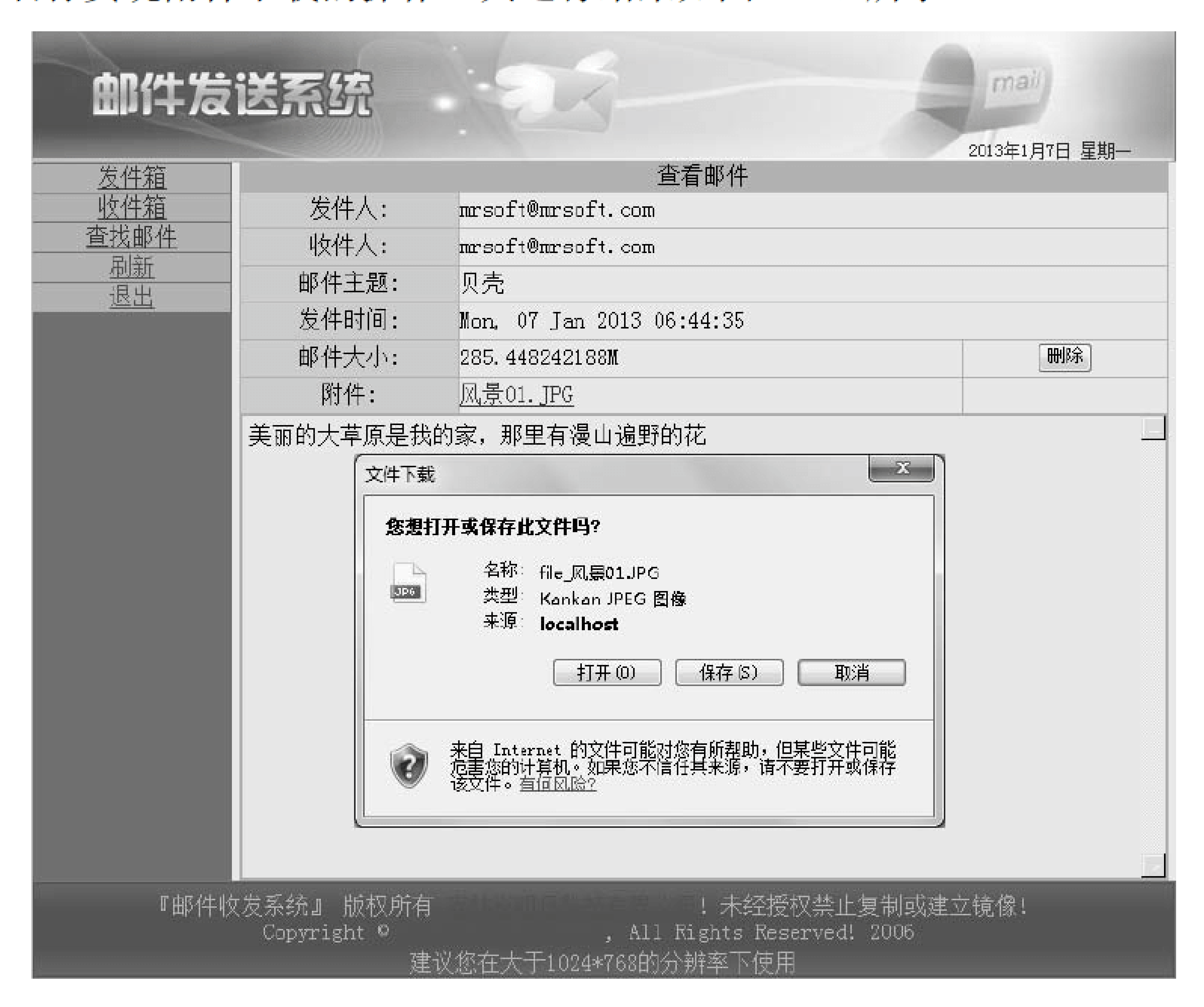


图 12.19 下载附件

在 down.php 文件中，根据超链接传递的附件名称，通过 header()函数完成附件的下载操作，其关键代码如下：

```
<?php
session_start();                         //初始化 Session 变量
$filename='file/'.$_GET['filename'];     //获取附件在服务器中的存储位置
if(!isset($_SESSION['user'])){           //判断当前用户的访问权限
echo "<script>alert('对不起，本站暂时停止该文件下载!');history.back();</script>";
exit;
}
$fp=fopen($filename,"r");                //打开指定的文件
header("Content-type:application/octet-stream");
header("Accept-ranges:bytes");
header("Accept-length:".filesize($filename));
header("Content-Disposition:attachment;filename=".$filename);
echo fread($fp,filesize($filename));     //输出文件
```



```
fclose($fp);                            //关闭文件
?>
```

12.5.6　删除邮件

在查看邮件详细内容页面中，单击“删除”按钮，将跳转到 deletemail.php 文件中，根据超链接传递的邮件 id 和附件名称，调用 POP3 类中的 dele()方法完成邮件的删除操作。如果存在附件，则应用 unlink()函数删除存储于服务器下的附件。deletemail.php 文件的代码如下：

```
<!—*******************************deletemail.php**********************************-->
<?php
session_start();
if(isset($_SESSION['user'])){           //判断当前用户的权限
include("mailclass.php");               //包含 POP3 发邮件类
$rec=new pop3($_SESSION['host'],110,10);           //对类进行初始化
$rec->open();                           //连接 POP3 服务器
//对当前用户进行身份验证
$rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']);
$rec->dele($_GET['del']);               //调用 POP3 类的 dele()方法来删除$_GET[del]指定的邮件
$rec->close();                          //关闭与 POP3 服务器的连接，并执行删除操作
if(!empty($_GET['filename'])){
unlink('file/'.$_GET['filename']);
}
echo "<script>alert('删除成功!');window.location.href='receivemail.php';</script>";
}else{
echo "<script>alert('您不具备删除权限，请正确登录!');window.location.href='index.php';;</script>";
}
?>
```

12.5.7　查找邮件

为了方便用户，该邮件系统提供了两种搜索方式，分别为按邮件主题进行查找和按发件人地址进行查找。查找邮件的操作在 findmail.php 文件中执行，页面如图 12.20 所示。



图 12.20　查找邮件页面

查找邮件创建的 form 表单如下：

```
<form name="form1" method="post" action="<?php echo $PHP_SELF;?>" onSubmit="return chkinput(this)">
<tr>
<td width="143"><div align="right">查找邮件:</div></td>
<td width="92"><div align="center">
<select name="findtype" class="inputcss">
```

```
<option selected value=1>邮件主题</option>
<option value=0>发件人</option>
</select>
</div></td>
<td width="191"><div align="center">
<input type="text" name="info" size="30" class="inputcss" style="background-color:#e8f4ff "
onMouseOver="this.style.backgroundColor='#ffffff'" onMouseOut="this.style.backgroundColor='#e8f4ff'">
</div></td>
<td width="224"><div align="left">
  <input type="submit" value="开始" class="buttoncss" name="submit">
</div></td>
</tr>
</form>
```

查找邮件的处理操作在 findmail.php 页中完成，其关键代码如下：

```
<!--*********************************findmail.php***********************************-->
……
<?php
if(isset($_POST['submit'])){                          //判断提交按钮值是否被设置
?>
<tr>
<td width="188" height="20" bgcolor="#E9E8E8"><div align="center">主题</div></td>
<td width="153" height="20" bgcolor="#E9E8E8"><div align="center">发件人</div></td>
<td width="209" height="20" bgcolor="#E9E8E8"><div align="center">时间</div></td>
<td width="97" height="20" bgcolor="#E9E8E8"><div align="center">大小</div></td>
</tr>
<?php
include("mailclass.php");                             //载入类文件
$rec=new pop3($_SESSION['host'],110,10);              //实例化类
$rec->open();                                         //连接服务器
$rec->login(substr($_SESSION['user'],0,strpos($_SESSION['user'],'@')),$_SESSION['pass']);  //用户登录
$rec->stat();                                         //调用 POP3 类的 stat()方法取得总邮件数和总邮件大小
$rec->listmail();                                     //调用 POP3 类的 listmail()方法取得每个邮件的大小及序号
if($_POST['findtype']==1){                            //判断查询的类型，1 表示根据邮件主题查询
$sum=0;                                               //用$sum 记录查找到邮件的总个数
for($i=1;$i<=$rec->messages;$i++){                    //通过循环遍历所有邮件
$rec->getmail($i);                                    //取得第$i 封邮件头及邮件内容
for($j=0;$j<count($rec->head);$j++){                  //通过循环获取邮件头所有内容
if(substr($rec->head[$j],0,7)=="Subject"){            //获取邮件主题
//去掉邮件主题的提示文字 Subject:
$head1=substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8);
}
}
if(preg_match("/$_POST[info]/i",trim($head1))){       //利用正则表达式进行匹配查找
$sum++;                                               //如果查找到匹配的主题，则使 sum 值加 1
?>
<tr>
<td height="20" bgcolor="#E9E8E8"><div align="left">
<a href="readmail.php?mailid=<?php echo $i;?>">
```



```
<?php
for($j=0;$j<count($rec->head);$j++){                    //显示邮件主题
if(substr($rec->head[$j],0,7)=="Subject"){
echo substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8)."<br>\n";
}
}
?>
</a>
</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){                    //显示发件人地址
if(substr($rec->head[$j],0,4)=="From"){
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
}
}
?>
</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){                    //显示发件时间
if(substr($rec->head[$j],0,4)=="Date"){
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";
}
}
?>
</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
$size=$rec->mail_list[$i]['size'];                      //显示邮件大小
if($size>=1024){
echo substr(($size/1024),0,4)."M";
}elseif($size>1024*1024){
echo substr(($size/(1024*1024)),0,4)."G";
}elseif($size<1024){
echo ($size)."KB";
}
?>
</div></td>
</tr>
<?php
}
$rec->head=NULL;
}
if($sum==0){                    //如果$sum 的值最终还为 0，则说明没有查找到匹配的邮件主题，这时给出提示
?>
<tr>
<td height="20" colspan="4" bgcolor="#E9E8E8"><div align="center">对不起，没有查找到您要找的邮件!</div>
</td>
</tr>
```

```
<?php
}
}elseif($_POST['findtype']==0){ //如果$_POST['findtype']的值为 0，则按发件人地址进行查找
$sum=0;
for($i=1;$i<=$rec->messages;$i++){
$rec->getmail($i);
for($j=0;$j<count($rec->head);$j++){
if(substr($rec->head[$j],0,4)=="From"){
$head1=substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5);
}
}
if(trim($head1)==trim($_POST['info'])){
$sum++;
?>
<tr>
<td height="20" bgcolor="#E9E8E8"><div align="left"><a href="readmail.php?mailid=<?php echo $i;?>">
<?php
for($j=0;$j<count($rec->head);$j++){
if(substr($rec->head[$j],0,7)=="Subject"){
echo substr(htmlspecialchars($rec->head[$j]),8,strlen(trim(htmlspecialchars($rec->head[$j])))-8)."<br>\n";
}
}
?>
</a></div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){
if(substr($rec->head[$j],0,4)=="From"){
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-5)."<br>\n";
}
}
?>
</div></td>
<td height="20" bgcolor="#E9E8E8"><div align="center">
<?php
for($j=0;$j<count($rec->head);$j++){
if(substr($rec->head[$j],0,4)=="Date"){
echo substr(htmlspecialchars($rec->head[$j]),5,strlen(trim(htmlspecialchars($rec->head[$j])))-10)."<br>\n";
}
}
?>
</div></td>
<td height="2" bgcolor="#E9E8E8"><div align="center">
<?php
$size=$rec->mail_list[$i]['size'];
if($size>=1024){
 echo substr(($size/1024),0,4)."M";
}elseif($size>1024*1024){
  echo substr(($size/(1024*1024)),0,4)."G";
```



```
}elseif($size<1024){
    echo ($size)."KB";
}
        ?>
</div></td>
</tr>
<?php
}
$rec->head=NULL;
}
if($sum==0){
?>
<tr>
<td height="20" colspan="4" bgcolor="#E9E8E8"><div align="center">对不起，没有查找到您要找的邮件!</div>
</td>
</tr>
        <?php
}
}
        }
        ?>
```

12.6　注 销 系 统

为了保护个人隐私，用户使用完本系统后应及时退出。本邮件发送系统完成注销系统功能的文件为 logout.php，其详细代码如下：

```
<!—*******************************logout.php*********************************-->
<?php
session_start();                                    //初始化 Session 变量
session_destroy();                                  //撤销所有 Session 变量
header("location:index.php");                       //重新定位到登录页面
?>
```

注销系统模块通过 session_destroy()函数撤销所有的 Session 变量，这样保存在 Session 变量中的所有与用户有关的信息也不再存在。

12.7　技 术 提 炼

12.7.1　利用 mail()函数发送电子邮件

PHP 中提供了可以直接发送电子邮件的 mail()函数。对于非商业用的邮件系统使用该函数最为方便，由于 mail()函数不能进行 SMTP 认证，所以最好安装一个 SMTP 服务器来转发邮件。mail()函数的



语法如下：

```
bool mail(string to, string subject, string message[ ,string additional_headers][ ,string additional_parameters])
```

mail()函数的参数说明如表 12.1 所示。

表 12.1　mail()函数的参数说明

参　　数	描　　述
to	收件人地址
subject	电子邮件的主题
message	电子邮件的内容
additional_headers	可选参数，用来将一些信息插入到 E-mail 的头部，如发送人信息、抄送地址、密送地址等。该参数如果有多个信息，信息之间用“\r\n”分隔
additional_parameters	其他邮件参数

注意 用 mail()函数发送邮件，需要在 php.ini 中进行 SMTP 邮件服务器地址和端口设置，否则邮件无法通过 SMTP 服务器发送出去。设置如下：

```
[mail function]
; For Win32 only.
SMTP = 192.168.1.42
smtp_port = 25
```

SMTP 是发送邮件的服务器地址。smtp_port 为 SMTP 服务器的端口号，一般默认为 25。

邮件发送系统是通过 mail()函数将电子邮件发送出去的，所以应该在系统中架设一个 SMTP 服务器。下面的代码为该邮件发送系统利用 mail()函数发送邮件的过程。

```
...
if(@mail($mailto, $mailsubject,$mailbody,$headers)) {            //通过 mail()函数发送邮件，并给出邮件发送结果
  echo "<script>alert('邮件发送成功！');history.back()</script>";
} else {
      echo "<script>alert('邮件未发送成功！');history.back()</script>";
}
```

☑ $mailto：表示收件人地址。
☑ $mailsubject：表示邮件主题。
☑ $mailbody：表示邮件内容。
☑ $headers：表示邮件头内容。

如果该邮件发送成功或失败系统将给出提示，并且用“@”屏蔽错误。

12.7.2　带附件的邮件发送

发送带附件的邮件并不是简单地将附件内容上传到 POP3 服务器的某个目录下，而是首先将附件





进行二进制编码，之后将该编码连接到邮件内容后面，当然邮件内容和附件编码之间需要用标记进行分隔，该标记在邮件头中定义。下面内容为带附件的邮件组织形式：

```
Received: from mrkj ([192.168.1.227]) by MRZTS with Microsoft SMTPSVC(6.0.3790.0);
Mon, 18 Dec 2006 09:02:40 +0800
Date: Mon, 18 Dec 2006 00:57:26 +0000
Subject: 12
To: zts***@163.com
Content-type: multipart/mixed; Boundary= 4585e7760018e
From:zts***@163.com
Return-Path: zts***@163.com
Message-ID:
X-OriginalArrivalTime: 18 Dec 2006 01:02:40.0187 (UTC) FILETIME=[372680B0:01C72240]
--4585e7760018
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: 8bit
 （此处为邮件内容）
--4585e7760018
Content-type: text/plain; name=shop.txt
Content-disposition: attachment; filename=shop.txt
Content-transfer-encoding: base64
wffV4rXEtcS1xLXEtcS1xLTvtb21xNK7ICC13LXctcS1xA== （此处为附件的 base64 编码）
--4585e7760018--
.
```

带附件的电子邮件的不同部分之间用分界线来分隔，分界线在 Content--type 头中定义，如 Boundary= 4585e7760018e，其中 4585e7760018e 表示不同部分的分界线。邮件的每个新部分以两个连字号（--）和分界线开始，最后一个分界线后也有两个连字号，表示该邮件中没有其他部分。在每个分界线后有一些行，用来告诉邮件程序这部分内容的类型，如上面例子中第一个分界线后面的以 Content-type:text/plain 开头的行，这些行说明后面的部分是 ISO-8859-1 字符集的纯文本形式，跟在第二个分界线后的行告诉邮件程序现在的部分是一个.txt 文件，文件名是 shop.txt。Content-disposition 这行告诉邮件程序如果可能就以内嵌的方式显示附件，现在新的邮件程序会在消息后显示.txt 文件的内容，如果 Content-disposition 被设为 attachment，那么邮件程序就不会显示.txt 文件的内容，而是显示一个连接到文件的图标，收件人要看附件的内容，必须单击这个图标。一般情况下，如果附件是一些文本，Content-disposition 会被设为 inline，这是因为现在大部分邮件程序能够不借助其他浏览器而直接显示附件（文本）的内容。如果附件不是文本（如图片或其他类似的内容），Content-disposition 就设为 attachment。

12.7.3　利用 fsockopen()函数接收电子邮件

上面介绍的 mail()函数只能发送邮件而无法接收，利用 fsockopen()函数除了可以发送邮件外，还可以接收邮件。fsockopen()函数的语法如下：

```
int fsockopen( string hostname, int port [, int errno [, string errstr [, float timeout]]])
```

fsockopen()函数的参数说明如表 12.2 所示。



表 12.2　fsockopen()函数的参数说明

参　　数	说　　明
hostname	在 Internet 中使用时，参数 hostname 及 port 分别代表网址及埠号。在 UNIX 情形下可做 IPC，hostname 参数表示到 socket 的路径，port 配置为 0
port	
errno	主要作为错误处理使用，可以省略
errstr	
timeout	表示在指定的时间内没有连接上就中断

利用 fsockopen()函数连接服务器后，将返回一个 socket 句柄，之后通过文件操作函数，如 fgets()、fputs()、feof()等接收消息或发送命令。本邮件系统将利用 fsockopen()函数接收 POP3 服务器邮箱中的内容。



fsockopen()函数可以与很多服务器建立连接，如 SMTP 服务器、POP3 服务器等。

邮件发送系统利用 fsockopen()函数连接 POP3 服务器，代码如下：

```
…
if(!$this->connection=@fsockopen($this->hostname,$this->port,&$err_no,&$err_str,$this->timeout)){
…
```

☑ $this->hostname：表示 POP3 服务器地址。
☑ $this->port：表示 POP3 服务器端口号。
☑ $err_no：表示系统连接 POP3 服务器发生错误时返回的错误号。
☑ $err_str：表示系统连接 POP3 服务器发生错误时返回的错误信息。
☑ $this->timeout：表示系统连接 POP3 服务器时最大超时时间。

12.7.4　封装邮件操作类

在邮件发送系统中，将收邮件的方法封装到 POP3 类，并且将其单独存储于 mailclass.php 中。下面对收邮件类 POP3 进行系统讲解。

（1）定义收邮件类中的数据成员。在收邮件类 POP3 中，首先定义类中的数据成员，其代码如下：

```
<!--******************************mailclass.php*********************************-->
class pop3{
public  $hostname="";             //POP3 主机名
public $port=110;                 //主机的 POP3 端口号，一般为 110
public $timeout=5;                //连接主机的最大超时时间
public $connection=0;             //保存与主机的连接
public $state="DISCONNECTED";     //保存用户登录状态
public $debug=0;                  //标识是否处于调试状态，如果是则输出调试信息
public $err_str='';               //如果出错，保存错误信息
public $err_no='';                //如果出错，保存错误号
public $resp;                     //临时保存服务器的响应信息
public $apop;                     //指示需要使用加密方式进行密码验证，一般服务器不需要
public $messages;                 //保存邮箱中的邮件总数
```

```
public $size;                   //邮箱中的邮件总大小
public $mail_list;              //一维数组，保存各个邮件的大小及其在 POP 服务器上的序号
public $head=array();           //保存邮件头内容
public $body=array();           //保存邮件内容
```

（2）定义收邮件类的构造函数__construct()，其代码如下：

```
<!--********************************mailclass.php**********************************-->
function __construct($server,$port=110,$time_out=5){
$this->hostname=$server;
$this->port=$port;
$this->timeout=$time_out;
return true;
}
```

注意　上述使用的是 PHP 5 中构造函数的定义方法，而在 PHP 4 中构造函数名应该与类名相同，当然 PHP 5 也兼容 PHP 4 中构造函数的定义方法。该构造函数的作用是为数据成员$hostname、$port、$timeout 初始化。

（3）定义 open()方法连接 POP3 服务器。在 open()方法中，通过 fsockopen()函数连接 POP3 服务器，并根据返回信息判断服务器是否连接成功，如果成功，则设置标记服务器的数据成员$state 的值为 AUTHORIZATION。代码如下：

```
<!—********************************mailclass.php**********************************-->
function open(){
if($this->hostname==""){                //判断服务器名是否为空，是则给出提示信息
$this->err_str="请输入主机名！";
return false;
}
if($this->debug)
echo "正在连接服务器！";
//利用 fsockopen()连接 POP3 服务器
if(!$this->connection=@fsockopen($this->hostname,$this->port,&$err_no,&$err_str,$this->timeout)){
$this->err_str="连接服务器失败，错误信息：".$err_str."错误号：".$err_no;
return false;
}else{
$this->getresp();                       //利用 getresp()方法处理服务器端回送的信息
if($this->debug)                        //显示返回信息
$this->outdebug($this->resp);
if(substr($this->resp,0,3)!="+OK"){//如果返回信息的前 3 个字符不为“+OK”说明服务器连接失败，并给出错误信息
$this->err_str="服务器回送无效信息；".$this->resp."请检查服务器是否正确!";
return false;
}
$this->state="AUTHORIZATION";
return true;
}
}
```

技巧 收邮件类判断某条命令发送是否成功的方法是：通过 substr()函数提取出返回信息的前 3 个字符，如果该字符为“+OK”，表示命令被成功执行，这是因为如果 POP3 中某条命令被成功执行，返回信息的前 3 个字符为“+OK”。下面是 fsockopen()函数连接成功时返回的信息：

```
+OK Microsoft Windows POP3 Service Version 1.0 <17073156@MRZTS> ready.
```

（4）使用 getresp()方法从 POP3 服务器接收命令。getresp()方法用于取得服务器端回送的信息，并对信息做如下处理：去掉最后的回车换行符后，将回送信息保存到数据成员$resp 中。getresp()方法的代码如下：

```
<!—******************************mailclass.php*********************************-->
function getresp(){
for($this->resp="";;){
if(feof($this->connection))
return false;
$this->resp.=fgets($this->connection,100);        //用 fgets()函数接收命令
$length=strlen($this->resp);                      //获取返回信息的长度
if($length>=2 && substr($this->resp,$length-2,2)=="\r\n"){
$this->resp=strtok($this->resp,"\r\n");           //去掉回车换行符
return true;
}
}
}
```

（5）使用 outdebug()方法对特殊字符进行转换。outdebug()方法的作用是将调试信息$messages 显示出来，并把一些特殊字符进行转换及在行尾加上
，这样可使输出的调试信息便于阅读和分析。outdebug()方法的代码如下：

```
function outdebug($message){
echo htmlspecialchars($message)."<br>\n";
}
```

（6）使用 command()方法向 POP3 服务器发送命令。command()方法可以接收以下 3 个参数。

☑ $command：发送给服务器的命令。

☑ $return_length：指定从服务器回送信息中取得多长作为命令回送标识。

☑ $return_code：指定标识的正确值。

如果服务器回送的第一个字符为“+”，则认为命令已正确执行，也可以用前面提到的“+OK”来判断。command()方法的代码如下：

```
<!--*******************************mailclass.php*********************************-->
function command($command,$return_length=1,$return_code='+'){
if($this->connection==0){                //如果$this->connection 的值为 0，说明还没有连接服务器
$this->err_str="没有连接上任何服务器，请检查网络连接!";
return false;
```



```
}
if($this->debug)                          //如果处于调试状态，则给出调试信息
$this->outdebug(">>> $command");
if(!fputs($this->connection,"$command\r\n")){  //如果发送命令失败，则给出信息
$this->err_str="无法发送命令".$command;
return false;
}else{
$this->getresp();                         //如果发送命令成功，则用 getresp()方法来接收返回的信息
if($this->debug)
$this->outdebug($this->resp);
if(substr($this->resp,0,$return_length)!=$return_code){ //如果返回的信息不是以“+”开头，说明返回的信息无效
$this->err_str=$command."命令，服务器返回无效：".$this->resp;
return false;
}else
return true;
}
}
```

（7）使用 login()方法对登录到 POP3 服务器上的用户进行身份验证。如果验证通过，则设置记录状态的数据成员$state 的值为 TRANSACTION。代码如下：

```
<!--******************************mailclass.php******************************-->
function login($user,$password){
if($this->state!="AUTHORIZATION"){ //标志状态的值不为 AUTHORIZATION，说明还没有连接上服务器
$this->err_str="还没有连接到服务器或状态不对!";
return false;
}
if(!$this->apop){                         //不需要加密方式进行密码验证
//向服务器发送 USER 命令，并用返回信息的前 3 个字符是否为“+OK”来验证命令是否发送成功
if(!$this->command("USER $user",3,"+OK"))
return false;
//向服务器发送 PASS 命令，并用返回信息的前 3 个字符是否为“+OK”来验证命令是否发送成功
if(!$this->command("PASS $password",3,"+OK"))
return false;
}else{                                    //需要加密方式进行密码验证
//向服务器发送 APOP 命令，以加密的方式进行身份验证
if(!$this->command("APOP $user".md5($this->greeting.$password),3,"+OK"))
return false;
}
$this->state="TRANSACTION";       //如果验证通过，则使记录状态的数据成员$state 的值为 TRANSACTION
return true;
}
```

用户成功登录服务器时返回的信息如下：

```
+OK
+OK User successfully logged on
```

上面代码中第一个“+OK”说明 USER 命令发送成功，第二个“+OK”说明 PASS 命令发送成功。

（8）stat()方法获取邮件总数量及总大小。stat()方法向服务器发送 STAT 命令，从而获得邮件总数量和总大小。代码如下：

```
<!--******************************mailclass.php******************************-->
function stat(){
if($this->state!="TRANSACTION"){                //判断用户是否已经成功登录
$this->err_str="还没有连接到服务器或没有登录成功！";
return false;
}
  if(!$this->command("STAT",3,"+OK"))           //向服务器发送 STAT 命令
return false;
else{
$this->resp=strtok($this->resp," ");            //取得“+OK”
$this->messages=strtok(" ");                    //取得邮件总数
$this->size=strtok(" ");                        //取得邮件总字节数
return true;
}
}
```

STAT 命令发送成功后返回的信息如下：

```
+OK 3 1504
```

上面代码中，+OK 说明命令发送成功；3 表示邮箱中有 3 个邮件；1504 表示邮件总大小为 1504 字节。

在 stat()方法中应用 strtok()函数对字符串进行分隔。strtok()函数的语法如下：

```
string strtok(string str1, string str2)
```

上述代码实现了从 str1 中返回 str2 首字母前的字符串，如果某程序中有多个 strtok()函数，只需要第一个 strtok()函数给出 str1 的值，第二个以后默认的 str1 的值为上次返回值后剩余的字符串。

（9）定义 listmail()方法获取指定邮件的大小，其代码如下：

```
<!--******************************mailclass.php******************************-->
function listmail($mess=NULL,$uni_id=NULL){
if($this->state!="TRANSACTION"){                //判断用户是否已经成功登录
$this->err_str="还没有连接到服务器或没有登录成功!";
return false;
}
if($uni_id)                                     //如果指定$uni_id，则使用 UIDL 命令
$command="UIDL";
else
$command="LIST";                                //如果没指定$uni_id，则使用 LIST 命令
if($mess)
$command.=$mess;                                //如果指定$mess，则使用$mess 所指定的命令
if(!$this->command($command,3,"+OK")){          //利用 command()方法发送命令
return false;
}else{
```

```
$i=0;
$this->mail_list=array();                       //定义二维数组，用来保存每个邮件的大小
$this->getresp();                               //获取服务器返回的信息
while($this->resp!="."){                        // “.” 为邮件结束的标志
$i++;
if($this->debug){
$this->outdebug($this->resp);
}
if($uni_id){
$this->mail_list[$i]['num']=strtok($this->resp," ");
        $this->mail_list[$i]['size']=strtok(" ");       //获取某个邮件的大小
}else{
$this->mail_list[$i]["num"]=intval(strtok($this->resp," "));
$this->mail_list[$i]["size"]=intval(strtok(" "));
}
$this->getresp();
}
return true;
}
}
```

（10）定义 getmail()方法获取邮件头及邮件内容，代码如下：

```
function getmail($num=1,$line=-1){
if($this->state!="TRANSACTION"){              //判断用户是否已经成功登录
$this->err_str="不能收邮件，还没有连接服务器或登录成功！";
return false;
}
if($line<0)                                   //如果$line<0，则返回所有的邮件内容
$command="RETR $num";
else                                          //如果$line>=0，则返回邮件前$num 行内容
$command="TOP $num $line";
if(!$this->command("$command",3,"+OK"))       //向服务器发送命令
return false;
else{
$this->getresp();                             //如果命令发送成功，则获取返回信息
$is_head=true;
while($this->resp!="."){
if($this->debug)
        $this->outdebug($this->resp);
if(substr($this->resp,0,1)==".")              // “.” 号为邮件结束的标志
        $this->resp=substr($this->resp,1,strlen($this->resp)-1);
if(trim($this->resp)=="")                     //邮件头与邮件内容以空行分隔
$is_head=false;
if($is_head)
$this->head[]=$this->resp;                    //将邮件头每一行的内容放到 head[]数组中
else
$this->body[]=$this->resp;              //将邮件内容的每一行放到 body[]数组中
$this->getresp();
```

```
}
return true;
}
}
```

（11）定义 dele()方法删除指定邮件，代码如下：

```
function dele($num){
if($this->state!="TRANSACTION"){                    //判断用户是否已经成功登录
$this->err_str="不能删除邮件";
return false;
}
if(!$num){
$this->err_str="删除的邮件参数不对！";
return false;
}
if($this->command("DELE $num",3,"+OK"))             //删除$num 指定的邮件
return true;
else
return false;
}
```

（12）定义 close()方法断开与服务器的连接，代码如下：

```
function close(){                                   // close()方法的作用是删除$num 指定的邮件
if($this->connection!=0){
if($this->state=="TRANSACTION")
$this->command("QUIT",3,"+OK");   //向服务器发送 QUIT 命令，从而断开与服务器的连接和执行 DELE 命令
fclose($this->connection);
$this->connection=NULL;
$this->state="DISCONNECTED";
}
}
}
```

12.8 本 章 小 结

本章 Zend Framework 通过 Zend_Mail 组件构建发送邮件系统，使每个用户可以以非常快速的方式与任何一个角落的网络用户联络，同时可以以任何方式发送邮件。充分利用 E-mail 功能可以极大地提高工作效率，减轻用户的工作负担。通过本章的学习，读者可以尝试构建自己的发送邮件系统。

